

# Laboratorio 1: Introducción a microcontroladores y manejo de GPIOs

Profesor: Marco Villalta, Estudiante: Sofía Villalta Jinesta B88565

## I. INTRODUCCIÓN

Este laboratorio busca introducir los conceptos de microcontroladores y el manejo de GPIOs. Se utilizó el microcontrolador PIC12f683 y se empleó simulIDE para desarrollar un simulador de dado simplificado, utilizando leds y un botón, así como otros componentes necesarios para el correcto funcionamiento.

El objetivo principal del laboratorio es introducir el manejo de GPIOs, la generación de números aleatorios y el flujo de desarrollo que se pide para las prácticas dirigidas. La idea es que cada vez que es presionado el botón se debe mostrar de forma aleatoria una de las caras de un dado representadas por los números de LEDs encendidos por un instante.

## II. NOTA TEÓRICA

### II-A. Características principales

El microcontrolador PIC12F683 es un componente esencial en la familia de microcontroladores PIC. Es compacto y versátil, se destaca porque es eficiente energéticamente y puede realizar una gran variedad de tareas de control y procesamiento. Estos microcontroladores tienden a usarse en equipos de comunicación, telefonía, en aplicaciones médicas, entre otros. Su capacidad es limitada por la que se utiliza para realizar tareas puntuales. Sus principales características son:

- **Arquitectura:** Este microcontrolador es un flash de 8 bits y 8 pines. Su procesador tiene arquitectura RISC, lo que quiere decir que su set de instrucciones es reducido y son monociclo, es eficiente y permite una ejecución rápida de las instrucciones.
- **Memoria:** tiene una capacidad de memoria de programa Flash de 3.5K, el PIC12F683 puede guardar las instrucciones y datos esenciales requeridos para su correcta operación. Esto habilita la ejecución eficiente de algoritmos y permite gestionar mejor los datos.
- **Conversión Analógica-Digital (ADC):** El microcontrolador cuenta con un convertidor analógico digital de 10 bits de resolución con 4 canales.
- **Pines de entrada/salida:** Tiene hasta 5 pines de entrada/salida que permiten interactuar con el mundo externo, permitiendo la conexión con otros componentes.
- **Módulos de Temporización:** Este microcontrolador dispone de temporizadores que se pueden ajustar para crear períodos de tiempo exactos, lo que resulta beneficioso para supervisar eventos y actividades en entornos de sistemas en tiempo real.

También cuenta con una corriente de operación de  $11\mu A$  a 32 KHz y 2.0v y de  $220\mu A$  con 4 MHz y a 2.0v. Este microcontrolador tiene 8 pines en total, de estos 2 corresponden a VDD y VSS que son sus alimentaciones, los 6 restantes se utilizan de entrada, salida. La figura 1 muestra el diagrama del microcontrolador y en la figura 2 su diagrama de bloques.[1]

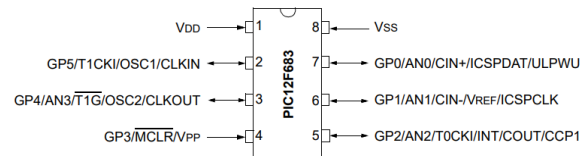


Figura 1. Diagrama del microcontrolador PIC12f683. [1]

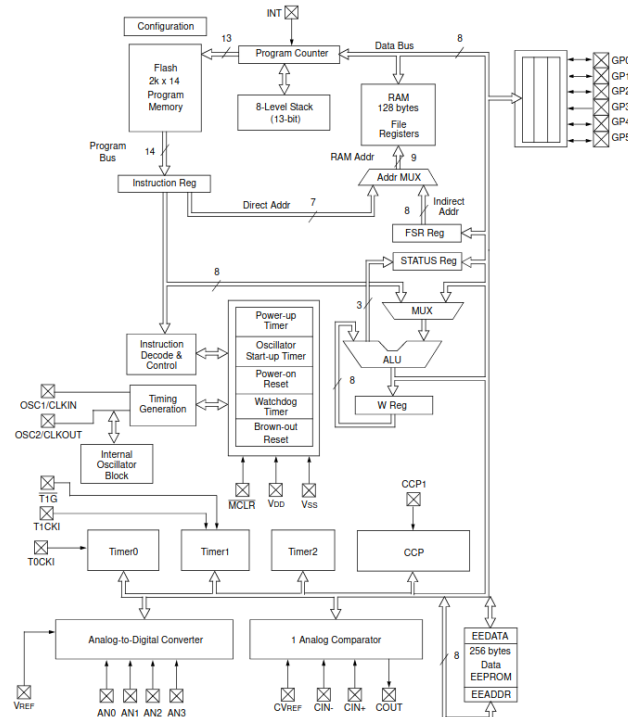


Figura 2. Diagrama de bloques del microcontrolador PIC12f683. [1]

*II-A1. Registros del PIC12F683:* Los registros importantes para la configuración de los pines de GPIOs son:

- **TRISIO:** Este registro determina la dirección de los datos (entrada o salida) para cada pin del GPIO. Si un bit en este registro se programa como '1', el correspondiente pin del GPIO se configura como entrada. Si se programa como '0', se configura como salida.
- **ANSEL:** Es el registro de Selección de Modo Analógico. Controla si los pines individuales del GPIO funcionan como entradas analógicas o como E/S digitales. Un bit configurado como '1' permite la entrada analógica, mientras que un '0' configura el pin para operaciones digitales.
- **CMCON:** Es el Registro de Control del Comparador. Controla la operación del comparador analógico interno del PIC12F683. Permite configurar el modo de operación del comparador y seleccionar qué pines se utilizan como entradas para el comparador.
- **OPTION:** Es un registro que configura varias funciones, incluyendo el prescaler de TMR0/WDT, la interrupción externa GP2/INT, TMR0 y las resistencias pull-up débiles en GPIO.
- **INTCON:** Es un registro que contiene bits de habilitación y bandera para cambios en GPIO y el pin externo GP2/INT.

### III. DISEÑO

La primera parte del diseño fue la del sistema del botón para generar los números aleatorios del 1 al 6 y que se mostrara el patrón en los leds. Se tuvo que tomar en cuenta el efecto rebote que se produce en el contacto de metal contra metal al estrifar un botón. Para corregir este efecto se implementó un capacitor de desacople de  $150\mu F$  para filtrar el ruido, como puede observarse en la figura 3 también se pueden observar las resistencias de pull up que se añadieron, estas son importantes cuando se emplea un botón en conjunto con un microcontrolador para resolver el problema de un estado incierto que puede surgir cuando el botón no está activado. Si el botón no se encuentra presionado, no se establece una conexión eléctrica directa entre la entrada del microcontrolador y el pin del botón, lo que podría llevar a que el pin permanezca en un estado flotante y ambiguo. En esta situación, el pin podría estar vulnerable al ruido eléctrico o interferencias, lo que potencialmente llevaría al microcontrolador a identificar un cambio de estado no deseado. Se escogieron resistencias con valores de  $120\Omega$  y un capacitor de  $150pF$  tomando de base la siguiente ecuación, que da un valor adecuado para el retardo:

$$\tau = R * C = (120\Omega + 120\Omega) + 150pF = 36ns \quad (1)$$

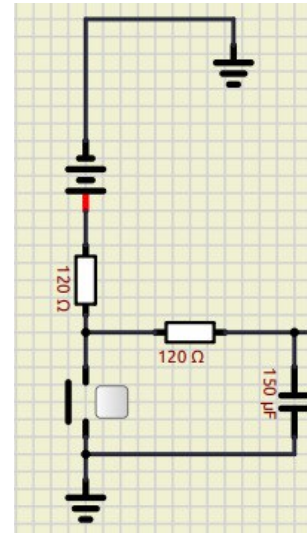


Figura 3. Primera etapa del diseño: manejo del botón.

La segunda etapa del diseño fue la de los leds y sus resistencias de protección, para esta parte se hizo el cálculo mediante la ley de Ohm sabiendo que la corriente máxima aceptada en las patillas del microcontrolador para un VDD de 5v es de 25mA, por lo que se utiliza la siguiente fórmula:

$$R = \frac{V}{I} = \frac{5V}{25mA} = 200\Omega \quad (2)$$

Las resistencias se ponen del lado del ánodo para proteger al diodo de la corriente, se conectó un led al pin P4, dos leds al pin P2 y tres al pin P1, el circuito completo se puede observar en la figura 4.

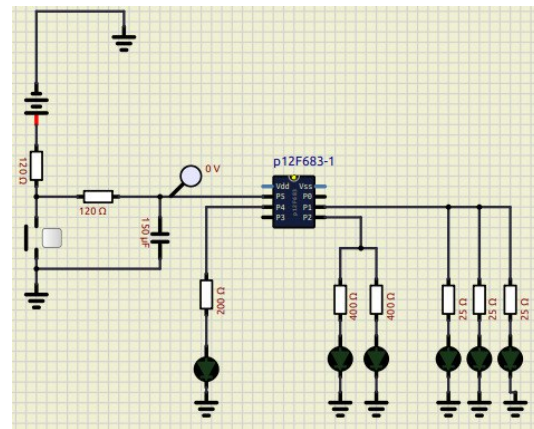


Figura 4. Segunda etapa del diseño: manejo de los leds.

La tabla de componentes y sus precios se puede apreciar en la tabla 1.

Componente	Valor	Cantidad
Resistencia de 120 ohms	\$5.99	1
Resistencia de 200 ohms	\$5.49	1
Resistencias de 400 ohms	\$5.49	2
Resistencias de 25 ohms	\$5.79	3
Microcontrolador PIC12F683	\$ 2.052	1
Paquete de diodos leds	\$6.99	1
Total	\$31,80	

Tabla I  
TABLA DE PRECIOS PARA COMPONENTES UTILIZADOS

#### IV. DESARROLLO Y ANÁLISIS DE RESULTADOS

Una vez explicada la metodología del diseño se procede a explicar el código implementado. Se utilizó el lenguaje de programación C. Primero se declararon dos funciones, la función delay que genera una pausa en la ejecución y la función generarAleatorio. Después de esto se definió la función principal o main, en ésta se configuran los pines del microcontrolador, el pin GP5 se configura como entrada pues corresponde al botón que se presiona para encender los LEDS, el programa verifica si el botón ha sido presionado ( $GP5 == 0$ ). Si es así, genera un valor aleatorio, enciende los LEDS correspondientes a ese valor, espera medio segundo y apaga todos los LEDS. Después se procedió a definir la función delay que crea una pausa en la ejecución del programa. Utiliza dos bucles anidados para esperar cierta cantidad de ciclos antes de continuar. Seguidamente se definió la función generarAleatorio que implementa un generador de números pseudoaleatorios utilizando un registro de desplazamiento con retroalimentación en serie (LFSR). En este caso, se asegura de generar valores entre 1 y 6, descartando valores mayores. En la figura 5 se puede observar un diagrama en pseudocódigo para entender mejor el funcionamiento del código de forma sencilla.

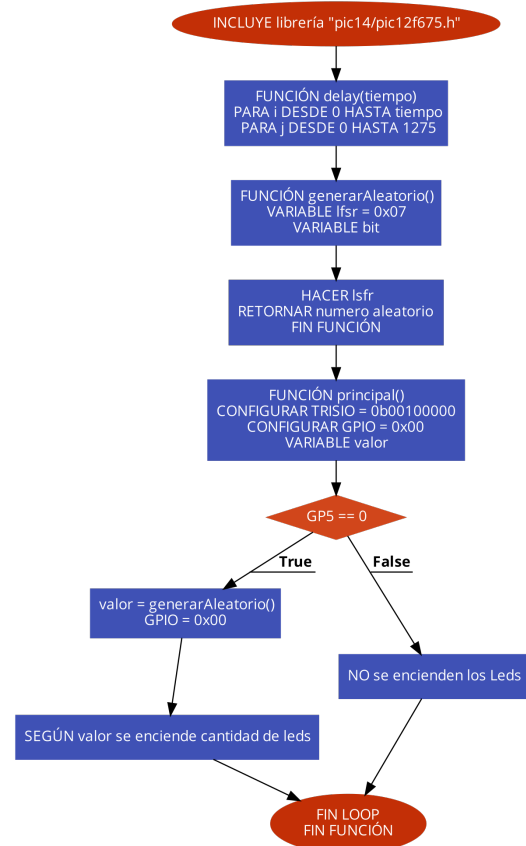


Figura 5. Diagrama del código

Después de realizar el código se procedió a hacer el Makefile y así poder probar el firmware en la simulación. Al inicio no se estaban obteniendo los resultados esperados pues se encendían todos los leds menos los correspondientes al número 6. Después de analizar a profundidad el código y el circuito se llegó a la conclusión de que había un problema con el pin GP3, este no se encendía inclusive probando códigos mucho más sencillos, se teoriza que es un problema de configuración y se solucionó reacomodando los pines, como se observa en la figura 4. Seguidamente se volvió a probar el funcionamiento y esta vez todos los números se encendían de forma aleatoria, en las figuras 6 y 8 se pueden observar dos ejemplos de su funcionamiento, cuando la cara corresponde a 1 y a 3, respectivamente.

#### V. CONCLUSIONES

- La conclusión principal es que este microcontrolador tiene aplicaciones reducidas pues cuenta con pocos pines, además si el algoritmo del código es muy complejo el microcontrolador no funciona, esta limitación no afectó tanto en este laboratorio pues se logró generar números aleatorios de manera sencilla pero es algo a tomar en cuenta.
- Se debe revisar bien la hoja del fabricante y sus pines, esto debido a que los pines pueden estar previamente configurados para otros usos. Dando problemas al ser utilizados.



- Es complicado generar números aleatorios sin bibliotecas externas pero se pueden utilizar métodos para simular la obtención de números aleatorios.

## VI. LINK A GIT

Este es el link del repositorio que se va a utilizar para el desarrollo de todos los laboratorios:

- [https://github.com/SofiVillalta29/Laboratorio\\_De\\_Microcontroladores/tree/main](https://github.com/SofiVillalta29/Laboratorio_De_Microcontroladores/tree/main)

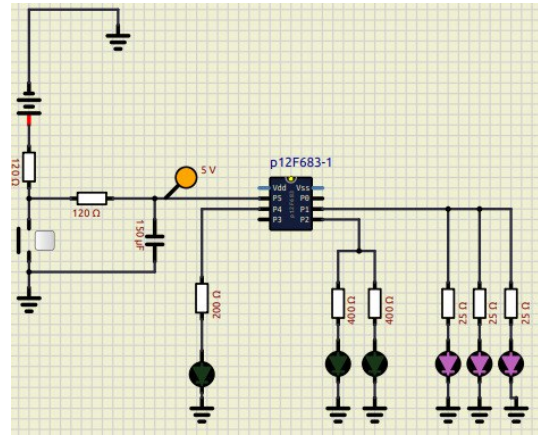


Figura 8. Circuito funcionando para número 3

## VII. APÉNDICES

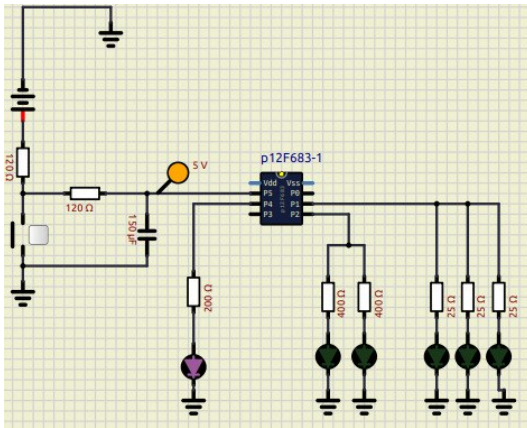


Figura 6. Circuito funcionando para número 1

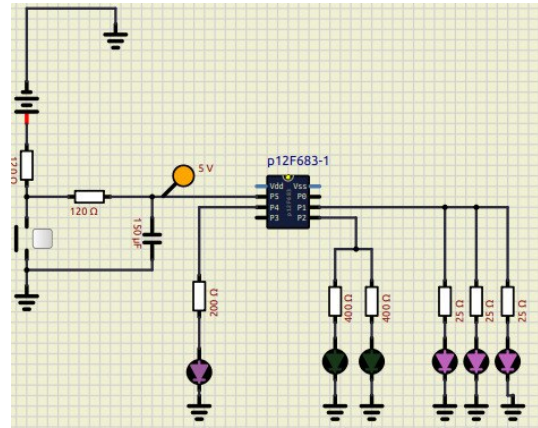


Figura 9. Circuito funcionando para número 4

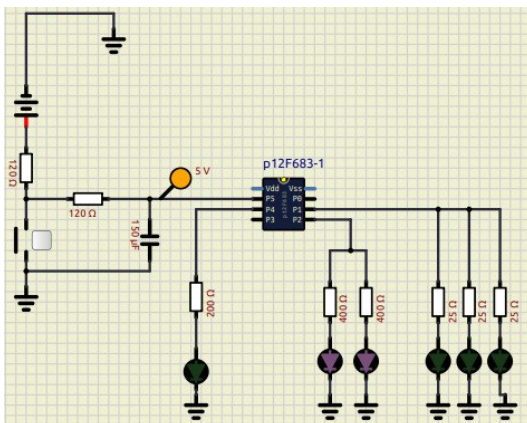


Figura 7. Circuito funcionando para número 2

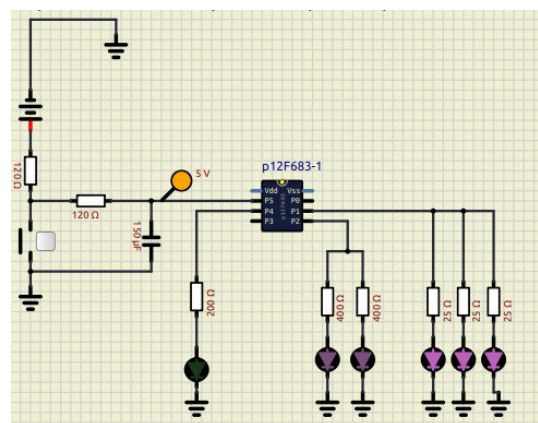


Figura 10. Circuito funcionando para número 5



# **PIC12F683**

## **Data Sheet**

8-Pin Flash-Based, 8-Bit  
CMOS Microcontrollers with  
nanoWatt Technology

---

**Note the following details of the code protection feature on Microchip devices:**

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

---

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights.

**Trademarks**

The Microchip name and logo, the Microchip logo, Accuron, dsPIC, KEELOQ, KEELOQ logo, microID, MPLAB, PIC, PICmicro, PICSTART, PRO MATE, PowerSmart, rfPIC, and SmartShunt are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.


AmpLab, FilterLab, Linear Active Thermistor, Migratable Memory, MXDEV, MXLAB, PS logo, SEEVAL, SmartSensor and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Application Maestro, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, ECAN, ECONOMONITOR, FanSense, FlexROM, fuzzyLAB, In-Circuit Serial Programming, ICSP, ICEPIC, Mindi, MiWi, MPASM, MPLAB Certified logo, MPLIB, MPLINK, PICkit, PICDEM, PICDEM.net, PICLAB, PICtail, PowerCal, PowerInfo, PowerMate, PowerTool, REAL ICE, rfLAB, rfPICDEM, Select Mode, Smart Serial, SmartTel, Total Endurance, UNI/O, WiperLock and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

All other trademarks mentioned herein are property of their respective companies.

© 2007, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

 Printed on recycled paper.

*Microchip received ISO/TS-16949:2002 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona, Gresham, Oregon and Mountain View, California. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.*

**QUALITY MANAGEMENT SYSTEM**  
**CERTIFIED BY DNV**  
**== ISO/TS 16949:2002 ==**

## 8-Pin Flash-Based, 8-Bit CMOS Microcontrollers with nanoWatt Technology

### High-Performance RISC CPU:

- Only 35 instructions to learn:
  - All single-cycle instructions except branches
- Operating speed:
  - DC – 20 MHz oscillator/clock input
  - DC – 200 ns instruction cycle
- Interrupt capability
- 8-level deep hardware stack
- Direct, Indirect and Relative Addressing modes

### Special Microcontroller Features:

- Precision Internal Oscillator:
  - Factory calibrated to  $\pm 1\%$ , typical
  - Software selectable frequency range of 8 MHz to 125 kHz
  - Software tunable
  - Two-Speed Start-up mode
  - Crystal fail detect for critical applications
  - Clock mode switching during operation for power savings
- Power-Saving Sleep mode
- Wide operating voltage range (2.0V-5.5V)
- Industrial and Extended temperature range
- Power-on Reset (POR)
- Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Brown-out Reset (BOR) with software control option
- Enhanced Low-Current Watchdog Timer (WDT) with on-chip oscillator (software selectable nominal 268 seconds with full prescaler) with software enable
- Multiplexed Master Clear with pull-up/input pin
- Programmable code protection
- High Endurance Flash/EEPROM cell:
  - 100,000 write Flash endurance
  - 1,000,000 write EEPROM endurance
  - Flash/Data EEPROM Retention: > 40 years

### Low-Power Features:

- Standby Current:
  - 50 nA @ 2.0V, typical
- Operating Current:
  - 11  $\mu$ A @ 32 kHz, 2.0V, typical
  - 220  $\mu$ A @ 4 MHz, 2.0V, typical
- Watchdog Timer Current:
  - 1  $\mu$ A @ 2.0V, typical

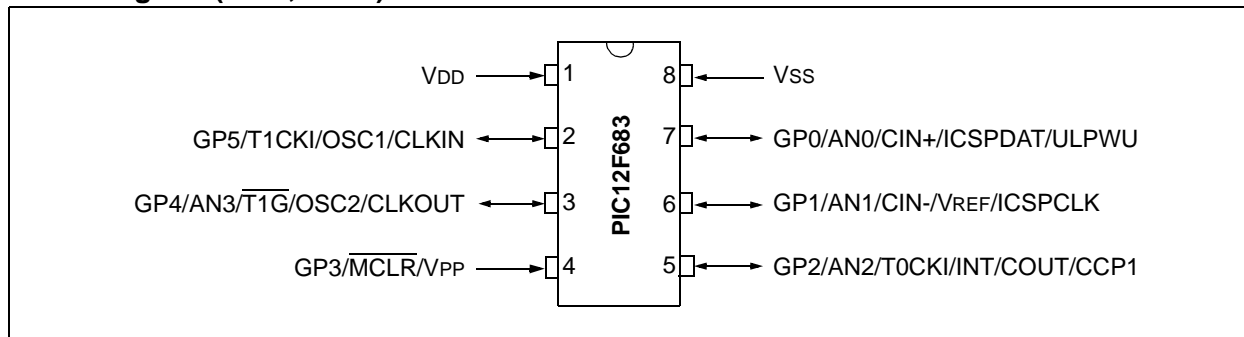
### Peripheral Features:

- 6 I/O pins with individual direction control:
  - High current source/sink for direct LED drive
  - Interrupt-on-pin change
  - Individually programmable weak pull-ups
  - Ultra Low-Power Wake-up on GP0
- Analog Comparator module with:
  - One analog comparator
  - Programmable on-chip voltage reference (CVREF) module (% of VDD)
  - Comparator inputs and output externally accessible
- A/D Converter:
  - 10-bit resolution and 4 channels
- Timer0: 8-bit timer/counter with 8-bit programmable prescaler
- Enhanced Timer1:
  - 16-bit timer/counter with prescaler
  - External Timer1 Gate (count enable)
  - Option to use OSC1 and OSC2 in LP mode as Timer1 oscillator if INTOSC mode selected
- Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler
- Capture, Compare, PWM module:
  - 16-bit Capture, max resolution 12.5 ns
  - Compare, max resolution 200 ns
  - 10-bit PWM, max frequency 20 kHz
- In-Circuit Serial Programming™ (ICSP™) via two pins

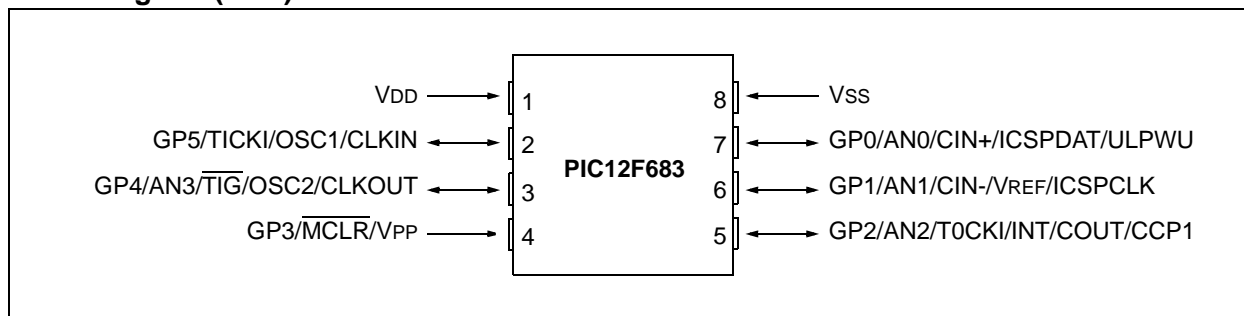
Device	Program Memory	Data Memory		I/O	10-bit A/D (ch)	Comparators	Timers 8/16-bit
	Flash (words)	SRAM (bytes)	EEPROM (bytes)				
PIC12F683	2048	128	256	6	4	1	2/1

# PIC12F683

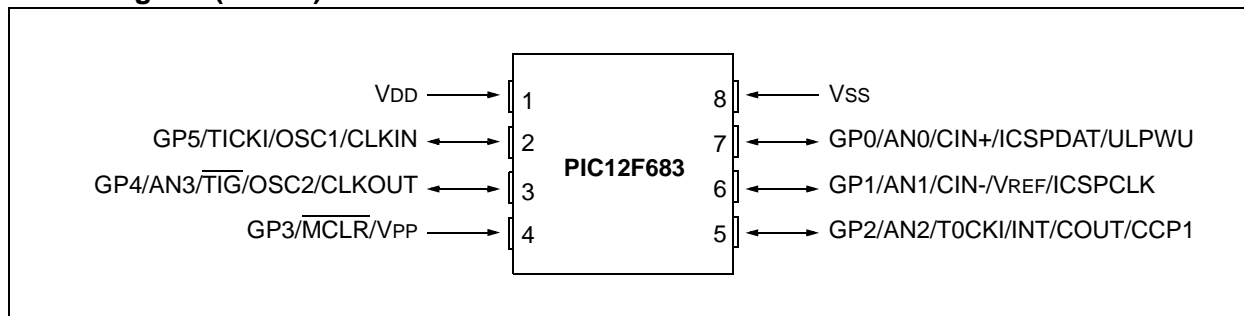
## 8-Pin Diagram (PDIP, SOIC)



## 8-Pin Diagram (DFN)



## 8-Pin Diagram (DFN-S)



**TABLE 1: 8-PIN SUMMARY**

I/O	Pin	Analog	Comparators	Timer	CCP	Interrupts	Pull-ups	Basic
GP0	7	AN0	CIN+	—	—	IOC	Y	ICSPDAT/ULPWU
GP1	6	AN1/VREF	CIN-	—	—	IOC	Y	ICSPCLK
GP2	5	AN2	COU	T0CKI	CCP1	INT/IOC	Y	—
GP3 <sup>(1)</sup>	4	—	—	—	—	IOC	Y <sup>(2)</sup>	$\overline{MCLR}$ /VPP
GP4	3	AN3	—	$\overline{T1G}$	—	IOC	Y	OSC2/CLKOUT
GP5	2	—	—	T1CKI	—	IOC	Y	OSC1/CLKIN
—	1	—	—	—	—	—	—	VDD
—	8	—	—	—	—	—	—	VSS

**Note 1:** Input only.

**2:** Only when pin is configured for external  $\overline{MCLR}$ .



## Table of Contents

1.0	Device Overview .....	5
2.0	Memory Organization .....	7
3.0	Oscillator Module (With Fail-Safe Clock Monitor).....	19
4.0	GPIO Port.....	31
5.0	Timer0 Module .....	41
6.0	Timer1 Module with Gate Control.....	44
7.0	Timer2 Module .....	49
8.0	Comparator Module.....	51
9.0	Analog-to-Digital Converter (ADC) Module .....	61
10.0	Data EEPROM Memory .....	71
11.0	Capture/Compare/PWM (CCP) Module .....	75
12.0	Special Features of the CPU .....	83
13.0	Instruction Set Summary .....	101
14.0	Development Support.....	111
15.0	Electrical Specifications.....	115
16.0	DC and AC Characteristics Graphs and Tables.....	137
17.0	Packaging Information.....	159
	Appendix A: Data Sheet Revision History.....	165
	Appendix B: Migrating From Other PIC® Devices .....	165
	The Microchip Web Site .....	171
	Customer Change Notification Service .....	171
	Customer Support.....	171
	Reader Response .....	172
	Product Identification System .....	173

## TO OUR VALUED CUSTOMERS

It is our intention to provide our valued customers with the best documentation possible to ensure successful use of your Microchip products. To this end, we will continue to improve our publications to better suit your needs. Our publications will be refined and enhanced as new volumes and updates are introduced.

If you have any questions or comments regarding this publication, please contact the Marketing Communications Department via E-mail at [docerrors@microchip.com](mailto:docerrors@microchip.com) or fax the **Reader Response Form** in the back of this data sheet to (480) 792-4150. We welcome your feedback.

### Most Current Data Sheet

To obtain the most up-to-date version of this data sheet, please register at our Worldwide Web site at:

<http://www.microchip.com>

You can determine the version of a data sheet by examining its literature number found on the bottom outside corner of any page. The last character of the literature number is the version number, (e.g., DS30000A is version A of document DS30000).

### Errata

An errata sheet, describing minor operational differences from the data sheet and recommended workarounds, may exist for current devices. As device/documentation issues become known to us, we will publish an errata sheet. The errata will specify the revision of silicon and revision of document to which it applies.

To determine if an errata sheet exists for a particular device, please check with one of the following:

- Microchip's Worldwide Web site; <http://www.microchip.com>
- Your local Microchip sales office (see last page)

When contacting a sales office, please specify which device, revision of silicon and data sheet (include literature number) you are using.

### Customer Notification System

Register on our web site at [www.microchip.com](http://www.microchip.com) to receive the most current information on all of our products.

# PIC12F683

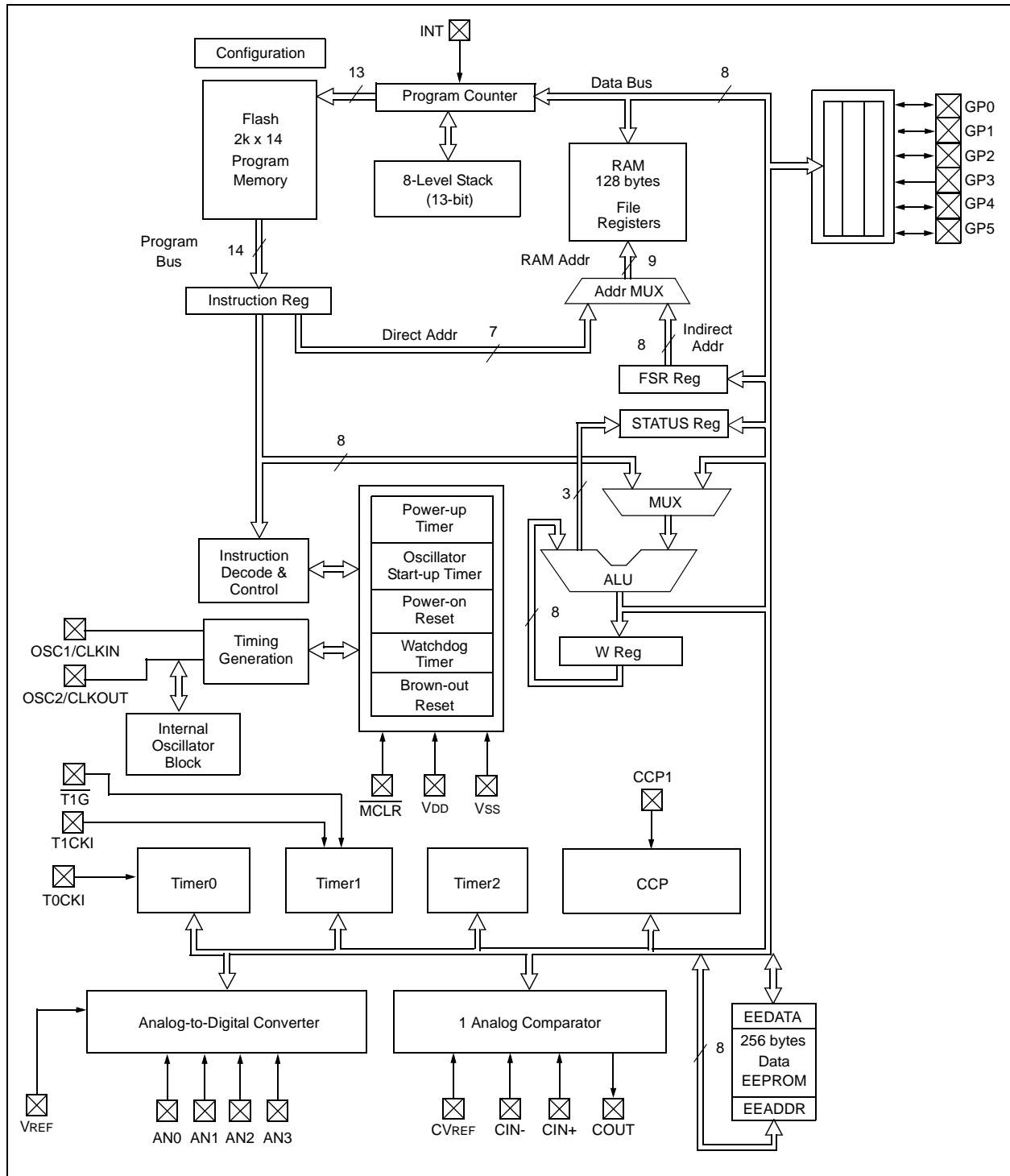
---

NOTES:

## 1.0 DEVICE OVERVIEW

The PIC12F683 is covered by this data sheet. It is available in 8-pin PDIP, SOIC and DFN-S packages. Figure 1-1 shows a block diagram of the PIC12F683 device. Table 1-1 shows the pinout description.

**FIGURE 1-1: PIC12F683 BLOCK DIAGRAM**



# PIC12F683

**TABLE 1-1: PIC12F683 PINOUT DESCRIPTION**

Name	Function	Input Type	Output Type	Description
VDD	VDD	Power	—	Positive supply
GP5/T1CKI/OSC1/CLKIN	GP5	TTL	CMOS	GPIO I/O with prog. pull-up and interrupt-on-change
	T1CKI	ST	—	Timer1 clock
	OSC1	XTAL	—	Crystal/Resonator
	CLKIN	ST	—	External clock input/RC oscillator connection
GP4/AN3/T1G/OSC2/CLKOUT	GP4	TTL	CMOS	GPIO I/O with prog. pull-up and interrupt-on-change
	AN3	AN	—	A/D Channel 3 input
	T1G	ST	—	Timer1 gate
	OSC2	—	XTAL	Crystal/Resonator
	CLKOUT	—	CMOS	Fosc/4 output
GP3/MCLR/VPP	GP3	TTL	—	GPIO input with interrupt-on-change
	MCLR	ST	—	Master Clear with internal pull-up
	VPP	HV	—	Programming voltage
GP2/AN2/T0CKI/INT/COU/CCP1	GP2	ST	CMOS	GPIO I/O with prog. pull-up and interrupt-on-change
	AN2	AN	—	A/D Channel 2 input
	T0CKI	ST	—	Timer0 clock input
	INT	ST	—	External Interrupt
	COU	—	CMOS	Comparator 1 output
	CCP1	ST	CMOS	Capture input/Compare output/PWM output
GP1/AN1/CIN-/VREF/ICSPCLK	GP1	TTL	CMOS	GPIO I/O with prog. pull-up and interrupt-on-change
	AN1	AN	—	A/D Channel 1 input
	CIN-	AN	—	Comparator 1 input
	VREF	AN	—	External Voltage Reference for A/D
	ICSPCLK	ST	—	Serial Programming Clock
GP0/AN0/CIN+/ICSPDAT/ULPWU	GP0	TTL	CMOS	GPIO I/O with prog. pull-up and interrupt-on-change
	AN0	AN	—	A/D Channel 0 input
	CIN+	AN	—	Comparator 1 input
	ICSPDAT	ST	CMOS	Serial Programming Data I/O
	ULPWU	AN	—	Ultra Low-Power Wake-up input
VSS	VSS	Power	—	Ground reference

**Legend:**

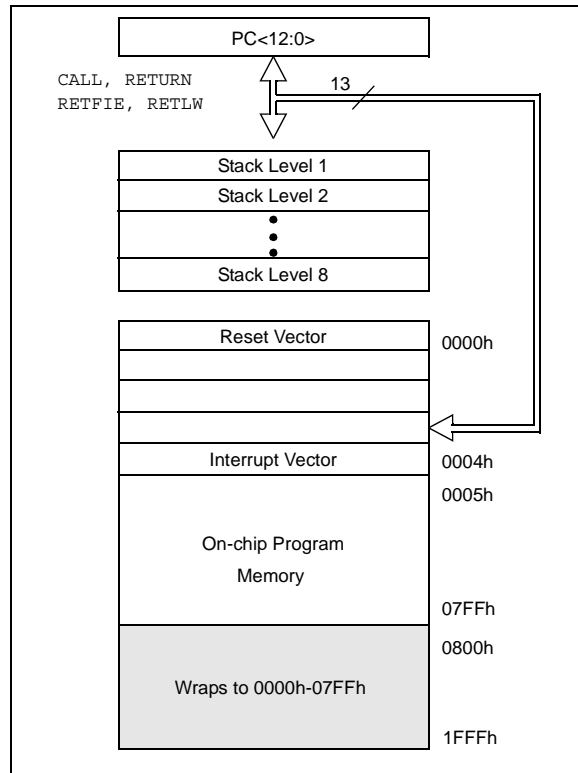
AN = Analog input or output	CMOS = CMOS compatible input or output
TTL = TTL compatible input	ST = Schmitt Trigger input with CMOS levels
HV = High Voltage	XTAL = Crystal

## 2.0 MEMORY ORGANIZATION

### 2.1 Program Memory Organization

The PIC12F683 has a 13-bit program counter capable of addressing an 8k x 14 program memory space. Only the first 2k x 14 (0000h-07FFh) for the PIC12F683 is physically implemented. Accessing a location above these boundaries will cause a wraparound within the first 2K x 14 space. The Reset vector is at 0000h and the interrupt vector is at 0004h (see Figure 2-1).

**FIGURE 2-1: PROGRAM MEMORY MAP AND STACK FOR THE PIC12F683**



### 2.2 Data Memory Organization

The data memory (see Figure 2-2) is partitioned into two banks, which contain the General Purpose Registers (GPR) and the Special Function Registers (SFR). The Special Function Registers are located in the first 32 locations of each bank. Register locations 20h-7Fh in Bank 0 and A0h-BFh in Bank 1 are General Purpose Registers, implemented as static RAM. Register locations F0h-FFh in Bank 1 point to addresses 70h-7Fh in Bank 0. All other RAM is unimplemented and returns '0' when read. RP0 of the STATUS register is the bank select bit.

#### RP0

- 0 → Bank 0 is selected
- 1 → Bank 1 is selected

**Note:** The IRP and RP1 bits of the STATUS register are reserved and should always be maintained as '0's.

\_\_\_\_\_

The special registers can be classified into two sets: core and peripheral. The Special Function Registers associated with the “core” are described in this section. Those related to the operation of the peripheral features are described in the section of that peripheral feature.

File Address		File Address	
Indirect addr. <sup>(1)</sup>	00h	Indirect addr. <sup>(1)</sup>	80h
TMR0	01h	OPTION_REG	81h
PCL	02h	PCL	82h
STATUS	03h	STATUS	83h
FSR	04h	FSR	84h
GPIO	05h	TRISIO	85h
	06h		86h
	07h		87h
	08h		88h
	09h		89h
PCLATH	0Ah	PCLATH	8Ah
INTCON	0Bh	INTCON	8Bh
PIR1	0Ch	PIE1	8Ch
	0Dh		8Dh
TMR1L	0Eh	PCON	8Eh
TMR1H	0Fh	OSCCON	8Fh
T1CON	10h	OSCTUNE	90h
TMR2	11h		91h
T2CON	12h	PR2	92h
CCPR1L	13h		93h
CCPR1H	14h		94h
CCP1CON	15h	WPU	95h
	16h	IOC	96h
	17h		97h
WDTCN	18h		98h
CMCON0	19h	VRCON	99h
CMCON1	1Ah	EEDAT	9Ah
	1Bh	EEADR	9Bh
	1Ch	EECON1	9Ch
	1Dh	EECON2 <sup>(1)</sup>	9Dh
ADRESH	1Eh	ADRESL	9Eh
ADCON0	1Fh	ANSEL	9Fh
General Purpose Registers 96 Bytes	20h	General Purpose Registers 32 Bytes	A0h
			BFh
			C0h
			EFh
			F0h
			FFh
	7Fh	Accesses 70h-7Fh	

BANK 0
BANK 1

Unimplemented data memory locations, read as '0'.

**Note 1:** Not a physical register.



**TABLE 2-1: PIC12F683 SPECIAL REGISTERS SUMMARY BANK 0**

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Page
Bank 0											
00h	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								xxxx xxxx	17, 90
01h	TMR0	Timer0 Module Register								xxxx xxxx	41, 90
02h	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	17, 90
03h	STATUS	IRP <sup>(1)</sup>	RP1 <sup>(1)</sup>	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	11, 90
04h	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	17, 90
05h	GPIO	—	—	GP5	GP4	GP3	GP2	GP1	GP0	--xx xxxx	31, 90
06h	—	Unimplemented								—	—
07h	—	Unimplemented								—	—
08h	—	Unimplemented								—	—
09h	—	Unimplemented								—	—
0Ah	PCLATH	—	—	—	Write Buffer for upper 5 bits of Program Counter				---	0 0000	17, 90
0Bh	INTCON	GIE	PEIE	TOIE	INTE	GPIE	TOIF	INTF	GPIF	0000 0000	13, 90
0Ch	PIR1	EEIF	ADIF	CCP1IF	—	CMIF	OSFIF	TMR2IF	TMR1IF	000- 0000	15, 90
0Dh	—	Unimplemented								—	—
0Eh	TMR1L	Holding Register for the Least Significant Byte of the 16-bit TMR1								xxxx xxxx	44, 90
0Fh	TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1								xxxx xxxx	44, 90
10h	T1CON	T1GINV	TMR1GE	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{T1SYNC}$	TMR1CS	TMR1ON	0000 0000	47, 90
11h	TMR2	Timer2 Module Register								0000 0000	49, 90
12h	T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	50, 90
13h	CCPR1L	Capture/Compare/PWM Register 1 Low Byte								xxxx xxxx	76, 90
14h	CCPR1H	Capture/Compare/PWM Register 1 High Byte								xxxx xxxx	76, 90
15h	CCP1CON	—	—	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	75, 90
16h	—	Unimplemented								—	—
17h	—	Unimplemented								—	—
18h	WDTCON	—	—	—	WDTPS3	WDTPS2	WDTPS1	WDTPS0	SWDTEN	---0 1000	97, 90
19h	CMCON0	—	COUT	—	CINV	CIS	CM2	CM1	CM0	-0-0 0000	56, 90
1Ah	CMCON1	—	—	—	—	—	—	T1GSS	CMSYNC	---- --10	57, 90
1Bh	—	Unimplemented								—	—
1Ch	—	Unimplemented								—	—
1Dh	—	Unimplemented								—	—
1Eh	ADRESH	Most Significant 8 bits of the left shifted A/D result or 2 bits of right shifted result								xxxx xxxx	61, 90
1Fh	ADCON0	ADFM	VCFG	—	—	CHS1	CHS0	GO/DONE	ADON	00-- 0000	65, 90

**Legend:** — = unimplemented locations read as '0', u = unchanged, x = unknown, q = value depends on condition,  
shaded = unimplemented

**Note 1:** IRP and RP1 bits are reserved, always maintain these bits clear.

# PIC12F683

**TABLE 2-2: PIC12F683 SPECIAL FUNCTION REGISTERS SUMMARY BANK 1**

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Page
Bank 1											
80h	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								xxxx xxxx	17, 90
81h	OPTION_REG	GPPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	12, 90
82h	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	17, 90
83h	STATUS	IRP <sup>(1)</sup>	RP1 <sup>(1)</sup>	RP0	T0	PD	Z	DC	C	0001 1xxx	11, 90
84h	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	17, 90
85h	TRISIO	—	—	TRISIO5	TRISIO4	TRISIO3	TRISIO2	TRISIO1	TRISIO0	--11 1111	32, 90
86h	—	Unimplemented								—	—
87h	—	Unimplemented								—	—
88h	—	Unimplemented								—	—
89h	—	Unimplemented								—	—
8Ah	PCLATH	—	—	—	Write Buffer for upper 5 bits of Program Counter				---	0 0000	17, 90
8Bh	INTCON	GIE	PEIE	T0IE	INTE	GPIE	T0IF	INTF	GPIF	0000 0000	13, 90
8Ch	PIE1	EEIE	ADIE	CCP1IE	—	CMIE	OSFIE	TMR2IE	TMR1IE	000- 0000	14, 90
8Dh	—	Unimplemented								—	—
8Eh	PCON	—	—	ULPWUE	SBOREN	—	—	POR	BOR	--01 --qq	16, 90
8Fh	OSCCON	—	IRCF2	IRCF1	IRCF0	OSTS <sup>(2)</sup>	HTS	LTS	SCS	-110 x000	20, 90
90h	OSCTUNE	—	—	—	TUN4	TUN3	TUN2	TUN1	TUN0	---0 0000	24, 90
91h	—	Unimplemented								—	—
92h	PR2	Timer2 Module Period Register								1111 1111	49, 90
93h	—	Unimplemented								—	—
94h	—	Unimplemented								—	—
95h	WPU <sup>(3)</sup>	—	—	WPU5	WPU4	—	WPU2	WPU1	WPU0	--11 -111	34, 90
96h	IOC	—	—	IOC5	IOC4	IOC3	IOC2	IOC1	IOC0	--00 0000	34, 90
97h	—	Unimplemented								—	—
98h	—	Unimplemented								—	—
99h	VRCON	VREN	—	VRR	—	VR3	VR2	VR1	VR0	0-0- 0000	58, 90
9Ah	EEDAT	EEDAT7	EEDAT6	EEDAT5	EEDAT4	EEDAT3	EEDAT2	EEDAT1	EEDAT0	0000 0000	71, 90
9Bh	EEADR	EEADR7	EEADR6	EEADR5	EEADR4	EEADR3	EEADR2	EEADR1	EEADR0	0000 0000	71, 90
9Ch	EECON1	—	—	—	—	WRERR	WREN	WR	RD	---- x000	72, 91
9Dh	EECON2	EEPROM Control Register 2 (not a physical register)								---- ----	72, 91
9Eh	ADRESL	Least Significant 2 bits of the left shifted result or 8 bits of the right shifted result								xxxx xxxx	66, 91
9Fh	ANSEL	—	ADCS2	ADCS1	ADCS0	ANS3	ANS2	ANS1	ANS0	-000 1111	33, 91

**Legend:** — = unimplemented locations read as '0', u = unchanged, x = unknown, q = value depends on condition, shaded = unimplemented

**Note 1:** IRP and RP1 bits are reserved, always maintain these bits clear.

**2:** OSTS bit of the OSCCON register reset to '0' with Dual Speed Start-up and LP, HS or XT selected as the oscillator.

**3:** GP3 pull-up is enabled when MCLRE is '1' in the Configuration Word register.

## 2.2.2.1 STATUS Register

The STATUS register, shown in Register 2-1, contains:

- Arithmetic status of the ALU
- Reset status
- Bank select bits for data memory (SRAM)

The STATUS register can be the destination for any instruction, like any other register. If the STATUS register is the destination for an instruction that affects the Z, DC or C bits, then the write to these three bits is disabled. These bits are set or cleared according to the device logic. Furthermore, the  $\overline{\text{TO}}$  and  $\overline{\text{PD}}$  bits are not writable. Therefore, the result of an instruction with the STATUS register as destination may be different than intended.

For example, `CLRF STATUS`, will clear the upper three bits and set the Z bit. This leaves the STATUS register as `000u u1uu` (where u = unchanged).

It is recommended, therefore, that only `BCF`, `BSF`, `SWAPF` and `MOVWF` instructions are used to alter the STATUS register, because these instructions do not affect any Status bits. For other instructions not affecting any Status bits, see the "Instruction Set Summary".

**Note 1:** Bits IRP and RP1 of the STATUS register are not used by the PIC12F683 and should be maintained as clear. Use of these bits is not recommended, since this may affect upward compatibility with future products.

**2:** The C and DC bits operate as a Borrow and Digit Borrow out bit, respectively, in subtraction.

### REGISTER 2-1: STATUS: STATUS REGISTER

Reserved	Reserved	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x
IRP	RP1	RP0	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C
bit 7							bit 0

#### Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

bit 7	<b>IRP:</b> This bit is reserved and should be maintained as '0'
bit 6	<b>RP1:</b> This bit is reserved and should be maintained as '0'
bit 5	<b>RP0:</b> Register Bank Select bit (used for direct addressing) 1 = Bank 1 (80h – FFh) 0 = Bank 0 (00h – 7Fh)
bit 4	<b><math>\overline{\text{TO}}</math>:</b> Time-out bit 1 = After power-up, <code>CLRWDT</code> instruction or <code>SLEEP</code> instruction 0 = A WDT time-out occurred
bit 3	<b><math>\overline{\text{PD}}</math>:</b> Power-down bit 1 = After power-up or by the <code>CLRWDT</code> instruction 0 = By execution of the <code>SLEEP</code> instruction
bit 2	<b>Z:</b> Zero bit 1 = The result of an arithmetic or logic operation is zero 0 = The result of an arithmetic or logic operation is not zero
bit 1	<b>DC:</b> Digit Carry/Borrow bit ( <code>ADDWF</code> , <code>ADDLW</code> , <code>SUBLW</code> , <code>SUBWF</code> instructions), For Borrow, the polarity is reversed. 1 = A carry-out from the 4th low-order bit of the result occurred 0 = No carry-out from the 4th low-order bit of the result
bit 0	<b>C:</b> Carry/Borrow bit <sup>(1)</sup> ( <code>ADDWF</code> , <code>ADDLW</code> , <code>SUBLW</code> , <code>SUBWF</code> instructions) 1 = A carry-out from the Most Significant bit of the result occurred 0 = No carry-out from the Most Significant bit of the result occurred

**Note 1:** For Borrow, the polarity is reversed. A subtraction is executed by adding the two's complement of the second operand. For rotate (`RRF`, `RLF`) instructions, this bit is loaded with either the high-order or low-order bit of the source register.

# PIC12F683

## 2.2.2.2 OPTION Register

The OPTION register is a readable and writable register, which contains various control bits to configure:

- TMR0/WDT prescaler
- External GP2/INT interrupt
- TMR0
- Weak pull-ups on GPIO

**Note:** To achieve a 1:1 prescaler assignment for Timer0, assign the prescaler to the WDT by setting PSA bit of the OPTION register to '1' See **Section 5.1.3 “Software Programmable Prescaler”**.

### REGISTER 2-2: OPTION\_REG: OPTION REGISTER

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
$\overline{\text{GPPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
bit 7							bit 0

#### Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

- bit 7  **$\overline{\text{GPPU}}$** : GPIO Pull-up Enable bit  
1 = GPIO pull-ups are disabled  
0 = GPIO pull-ups are enabled by individual PORT latch values in WPU register
- bit 6 **INTEDG**: Interrupt Edge Select bit  
1 = Interrupt on rising edge of INT pin  
0 = Interrupt on falling edge of INT pin
- bit 5 **T0CS**: Timer0 Clock Source Select bit  
1 = Transition on T0CKI pin  
0 = Internal instruction cycle clock ( $F_{\text{osc}}/4$ )
- bit 4 **T0SE**: Timer0 Source Edge Select bit  
1 = Increment on high-to-low transition on T0CKI pin  
0 = Increment on low-to-high transition on T0CKI pin
- bit 3 **PSA**: Prescaler Assignment bit  
1 = Prescaler is assigned to the WDT  
0 = Prescaler is assigned to the Timer0 module
- bit 2-0 **PS<2:0>**: Prescaler Rate Select bits

BIT VALUE	TIMER0 RATE	WDT RATE
000	1 : 2	1 : 1
001	1 : 4	1 : 2
010	1 : 8	1 : 4
011	1 : 16	1 : 8
100	1 : 32	1 : 16
101	1 : 64	1 : 32
110	1 : 128	1 : 64
111	1 : 256	1 : 128

**Note 1:** A dedicated 16-bit WDT postscaler is available. See **Section 12.6 “Watchdog Timer (WDT)”** for more information.

## 2.2.2.3 INTCON Register

The INTCON register is a readable and writable register, which contains the various enable and flag bits for TMR0 register overflow, GPIO change and external GP2/INT pin interrupts.

**Note:** Interrupt flag bits are set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the global enable bit, GIE of the INTCON register. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

### REGISTER 2-3: INTCON: INTERRUPT CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
GIE	PEIE	TOIE	INTE	GPIE	T0IF	INTF	GPIF
bit 7							bit 0

#### Legend:

R = Readable bit      W = Writable bit      U = Unimplemented bit, read as '0'  
 -n = Value at POR      '1' = Bit is set      '0' = Bit is cleared      x = Bit is unknown

- bit 7      **GIE:** Global Interrupt Enable bit  
             1 = Enables all unmasked interrupts  
             0 = Disables all interrupts
- bit 6      **PEIE:** Peripheral Interrupt Enable bit  
             1 = Enables all unmasked peripheral interrupts  
             0 = Disables all peripheral interrupts
- bit 5      **TOIE:** Timer0 Overflow Interrupt Enable bit  
             1 = Enables the Timer0 interrupt  
             0 = Disables the Timer0 interrupt
- bit 4      **INTE:** GP2/INT External Interrupt Enable bit  
             1 = Enables the GP2/INT external interrupt  
             0 = Disables the GP2/INT external interrupt
- bit 3      **GPIE:** GPIO Change Interrupt Enable bit<sup>(1)</sup>  
             1 = Enables the GPIO change interrupt  
             0 = Disables the GPIO change interrupt
- bit 2      **T0IF:** Timer0 Overflow Interrupt Flag bit<sup>(2)</sup>  
             1 = Timer0 register has overflowed (must be cleared in software)  
             0 = Timer0 register did not overflow
- bit 1      **INTF:** GP2/INT External Interrupt Flag bit  
             1 = The GP2/INT external interrupt occurred (must be cleared in software)  
             0 = The GP2/INT external interrupt did not occur
- bit 0      **GPIF:** GPIO Change Interrupt Flag bit  
             1 = When at least one of the GPIO <5:0> pins changed state (must be cleared in software)  
             0 = None of the GPIO <5:0> pins have changed state

**Note 1:** IOC register must also be enabled.

**2:** T0IF bit is set when TMR0 rolls over. TMR0 is unchanged on Reset and should be initialized before clearing T0IF bit.

# PIC12F683

## 2.2.2.4 PIE1 Register

The PIE1 register contains the interrupt enable bits, as shown in Register 2-4.

**Note:** Bit PEIE of the INTCON register must be set to enable any peripheral interrupt.

### REGISTER 2-4: PIE1: PERIPHERAL INTERRUPT ENABLE REGISTER 1

R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
EEIE	ADIE	CCP1IE	—	CMIE	OSFIE	TMR2IE	TMR1IE
bit 7							bit 0

#### Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

- bit 7      **EEIE:** EE Write Complete Interrupt Enable bit  
1 = Enables the EE write complete interrupt  
0 = Disables the EE write complete interrupt
- bit 6      **ADIE:** A/D Converter (ADC) Interrupt Enable bit  
1 = Enables the ADC interrupt  
0 = Disables the ADC interrupt
- bit 5      **CCP1IE:** CCP1 Interrupt Enable bit  
1 = Enables the CCP1 interrupt  
0 = Disables the CCP1 interrupt
- bit 4      **Unimplemented:** Read as '0'
- bit 3      **CMIE:** Comparator Interrupt Enable bit  
1 = Enables the Comparator 1 interrupt  
0 = Disables the Comparator 1 interrupt
- bit 2      **OSFIE:** Oscillator Fail Interrupt Enable bit  
1 = Enables the oscillator fail interrupt  
0 = Disables the oscillator fail interrupt
- bit 1      **TMR2IE:** Timer2 to PR2 Match Interrupt Enable bit  
1 = Enables the Timer2 to PR2 match interrupt  
0 = Disables the Timer2 to PR2 match interrupt
- bit 0      **TMR1IE:** Timer1 Overflow Interrupt Enable bit  
1 = Enables the Timer1 overflow interrupt  
0 = Disables the Timer1 overflow interrupt



## 2.2.2.5 PIR1 Register

The PIR1 register contains the interrupt flag bits, as shown in Register 2-5.

**Note:** Interrupt flag bits are set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the global enable bit, GIE of the INTCON register. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

**REGISTER 2-5: PIR1: PERIPHERAL INTERRUPT REQUEST REGISTER 1**

R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
EEIF	ADIF	CCP1IF	—	CMIF	OSFIF	TMR2IF	TMR1IF
bit 7							bit 0

**Legend:**

R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared
		x = Bit is unknown

- bit 7 **EEIF:** EEPROM Write Operation Interrupt Flag bit  
 1 = The write operation completed (must be cleared in software)  
 0 = The write operation has not completed or has not been started
- bit 6 **ADIF:** A/D Interrupt Flag bit  
 1 = A/D conversion complete  
 0 = A/D conversion has not completed or has not been started
- bit 5 **CCP1IF:** CCP1 Interrupt Flag bit  
Capture mode:  
 1 = A TMR1 register capture occurred (must be cleared in software)  
 0 = No TMR1 register capture occurred  
Compare mode:  
 1 = A TMR1 register compare match occurred (must be cleared in software)  
 0 = No TMR1 register compare match occurred  
PWM mode:  
 Unused in this mode
- bit 4 **Unimplemented:** Read as '0'
- bit 3 **CMIF:** Comparator Interrupt Flag bit  
 1 = Comparator 1 output has changed (must be cleared in software)  
 0 = Comparator 1 output has not changed
- bit 2 **OSFIF:** Oscillator Fail Interrupt Flag bit  
 1 = System oscillator failed, clock input has changed to INTOSC (must be cleared in software)  
 0 = System clock operating
- bit 1 **TMR2IF:** Timer2 to PR2 Match Interrupt Flag bit  
 1 = Timer2 to PR2 match occurred (must be cleared in software)  
 0 = Timer2 to PR2 match has not occurred
- bit 0 **TMR1IF:** Timer1 Overflow Interrupt Flag bit  
 1 = Timer1 register overflowed (must be cleared in software)  
 0 = Timer1 has not overflowed

# PIC12F683

## 2.2.2.6 PCON Register

The Power Control (PCON) register contains flag bits (see Table 12-2) to differentiate between a:

- Power-on Reset ( $\overline{\text{POR}}$ )
- Brown-out Reset ( $\overline{\text{BOR}}$ )
- Watchdog Timer Reset (WDT)
- External MCLR Reset

The PCON register also controls the Ultra Low-Power Wake-up and software enable of the  $\overline{\text{BOR}}$ .

The PCON register bits are shown in Register 2-6.

### REGISTER 2-6: PCON: POWER CONTROL REGISTER

U-0	U-0	R/W-0	R/W-1	U-0	U-0	R/W-0	R/W-x
—	—	ULPWUE	SBOREN	—	—	$\overline{\text{POR}}$	$\overline{\text{BOR}}$
bit 7							bit 0

#### Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

bit 7-6 **Unimplemented:** Read as '0'

bit 5 **ULPWUE:** Ultra Low-Power Wake-Up Enable bit

1 = Ultra Low-Power Wake-up enabled

0 = Ultra Low-Power Wake-up disabled

bit 4 **SBOREN:** Software BOR Enable bit<sup>(1)</sup>

1 = BOR enabled

0 = BOR disabled

bit 3-2 **Unimplemented:** Read as '0'

bit 1 **POR:** Power-on Reset Status bit

1 = No Power-on Reset occurred

0 = A Power-on Reset occurred (must be set in software after a Power-on Reset occurs)

bit 0 **BOR:** Brown-out Reset Status bit

1 = No Brown-out Reset occurred

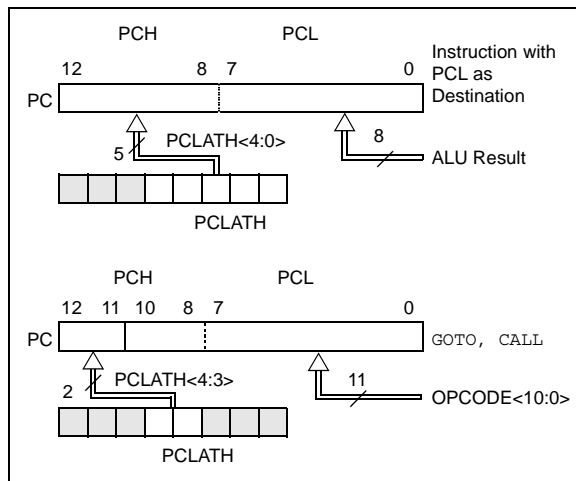
0 = A Brown-out Reset occurred (must be set in software after a Power-on Reset or Brown-out Reset occurs)

**Note 1:** Set BOREN<1:0> = 01 in the Configuration Word register for this bit to control the  $\overline{\text{BOR}}$ .

## 2.3 PCL and PCLATH

The Program Counter (PC) is 13 bits wide. The low byte comes from the PCL register, which is a readable and writable register. The high byte (PC<12:8>) is not directly readable or writable and comes from PCLATH. On any Reset, the PC is cleared. Figure 2-3 shows the two situations for the loading of the PC. The upper example in Figure 2-3 shows how the PC is loaded on a write to PCL (PCLATH<4:0> → PCH). The lower example in Figure 2-3 shows how the PC is loaded during a CALL or GOTO instruction (PCLATH<4:3> → PCH).

**FIGURE 2-3: LOADING OF PC IN DIFFERENT SITUATIONS**



### 2.3.1 COMPUTED GOTO

A computed GOTO is accomplished by adding an offset to the program counter (ADDWF PCL). When performing a table read using a computed GOTO method, care should be exercised if the table location crosses a PCL memory boundary (each 256-byte block). Refer to the Application Note AN556, "Implementing a Table Read" (DS00556).

### 2.3.2 STACK

The PIC12F683 family has an 8-level x 13-bit wide hardware stack (see Figure 2-1). The stack space is not part of either program or data space and the Stack Pointer is not readable or writable. The PC is PUSHed onto the stack when a CALL instruction is executed or an interrupt causes a branch. The stack is POPed in the event of a RETURN, RETLW or a RETFIE instruction execution. PCLATH is not affected by a PUSH or POP operation.

The stack operates as a circular buffer. This means that after the stack has been PUSHed eight times, the ninth push overwrites the value that was stored from the first push. The tenth push overwrites the second push (and so on).

**Note 1:** There are no Status bits to indicate stack overflow or stack underflow conditions.

**2:** There are no instructions/mnemonics called PUSH or POP. These are actions that occur from the execution of the CALL, RETURN, RETLW and RETFIE instructions or the vectoring to an interrupt address.

## 2.4 Indirect Addressing, INDF and FSR Registers

The INDF register is not a physical register. Addressing the INDF register will cause indirect addressing.

Indirect addressing is possible by using the INDF register. Any instruction using the INDF register actually accesses data pointed to by the File Select Register (FSR). Reading INDF itself indirectly will produce 00h. Writing to the INDF register indirectly results in a no operation (although Status bits may be affected). An effective 9-bit address is obtained by concatenating the 8-bit FSR register and the IRP bit of the STATUS register, as shown in Figure 2-4.

A simple program to clear RAM location 20h-2Fh using indirect addressing is shown in Example 2-1.

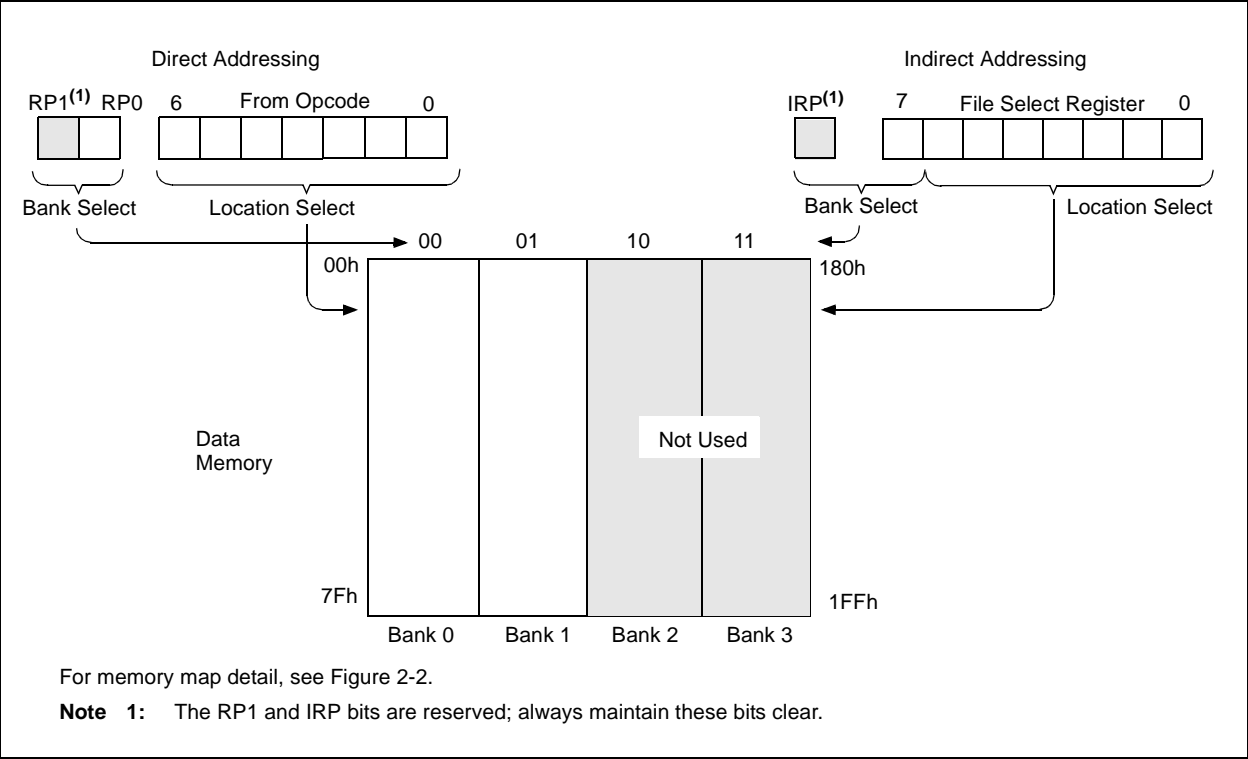
### EXAMPLE 2-1: INDIRECT ADDRESSING

```

        MOVLW    0x20    ;initialize pointer
        MOVWF    FSR      ;to RAM
NEXT    CLRF     INDF     ;clear INDF register
        INCF     FSR      ;inc pointer
        BTFSS    FSR,4    ;all done?
        GOTO     NEXT     ;no clear next
CONTINUE                                     ;yes continue
    
```

# PIC12F683

FIGURE 2-4: DIRECT/INDIRECT ADDRESSING PIC12F683



## 3.0 OSCILLATOR MODULE (WITH FAIL-SAFE CLOCK MONITOR)

### 3.1 Overview

The Oscillator module has a wide variety of clock sources and selection features that allow it to be used in a wide range of applications while maximizing performance and minimizing power consumption. Figure 3-1 illustrates a block diagram of the Oscillator module.

Clock sources can be configured from external oscillators, quartz crystal resonators, ceramic resonators and Resistor-Capacitor (RC) circuits. In addition, the system clock source can be configured from one of two internal oscillators, with a choice of speeds selectable via software. Additional clock features include:

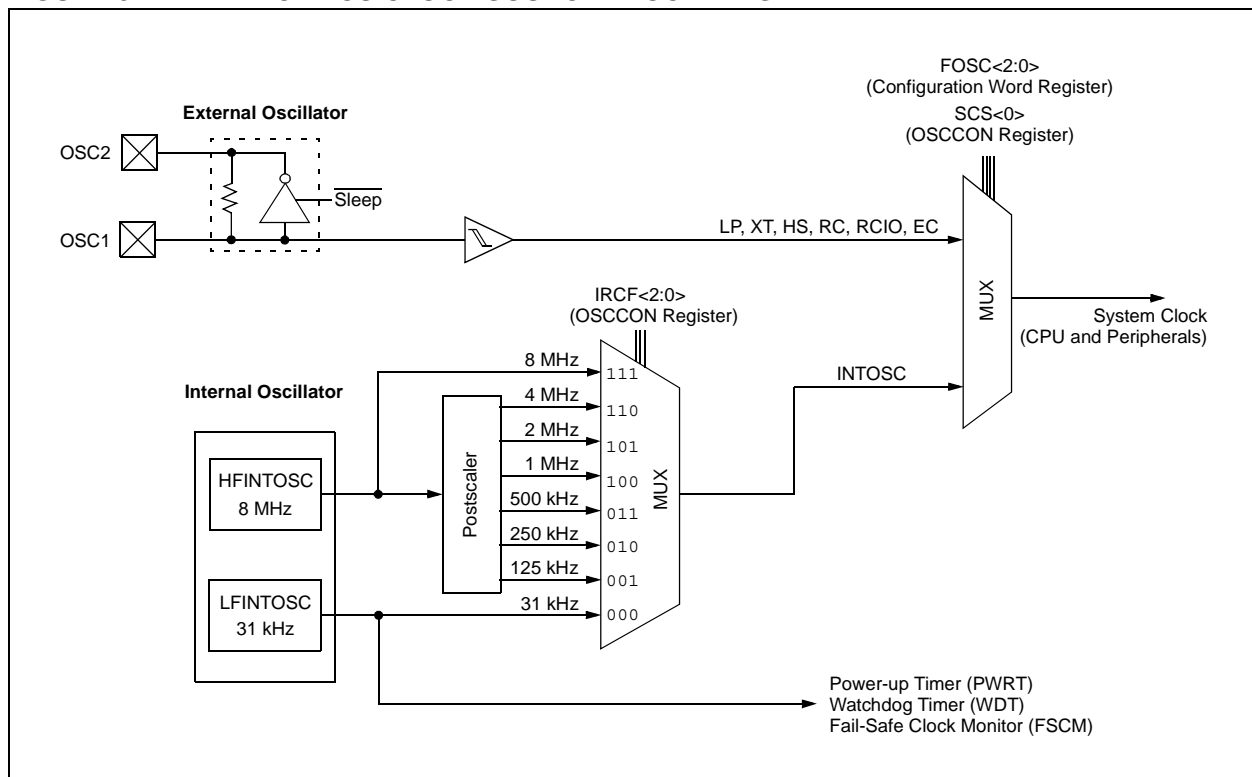
- Selectable system clock source between external or internal via software.
- Two-Speed Start-up mode, which minimizes latency between external oscillator start-up and code execution.
- Fail-Safe Clock Monitor (FSCM) designed to detect a failure of the external clock source (LP, XT, HS, EC or RC modes) and switch automatically to the internal oscillator.

The Oscillator module can be configured in one of eight clock modes.

1. EC – External clock with I/O on OSC2/CLKOUT.
2. LP – 32 kHz Low-Power Crystal mode.
3. XT – Medium Gain Crystal or Ceramic Resonator Oscillator mode.
4. HS – High Gain Crystal or Ceramic Resonator mode.
5. RC – External Resistor-Capacitor (RC) with Fosc/4 output on OSC2/CLKOUT.
6. RCIO – External Resistor-Capacitor (RC) with I/O on OSC2/CLKOUT.
7. INTOSC – Internal oscillator with Fosc/4 output on OSC2 and I/O on OSC1/CLKIN.
8. INTOSCIO – Internal oscillator with I/O on OSC1/CLKIN and OSC2/CLKOUT.

Clock Source modes are configured by the FOSC<2:0> bits in the Configuration Word register (CONFIG). The internal clock can be generated from two internal oscillators. The HFINTOSC is a calibrated high-frequency oscillator. The LFINTOSC is an uncalibrated low-frequency oscillator.

**FIGURE 3-1: PIC® MCU CLOCK SOURCE BLOCK DIAGRAM**



# PIC12F683

## 3.2 Oscillator Control

The Oscillator Control (OSCCON) register (Figure 3-1) controls the system clock and frequency selection options. The OSCCON register contains the following bits:

- Frequency selection bits (IRCF)
- Frequency Status bits (HTS, LTS)
- System clock control bits (OSTS, SCS)

**REGISTER 3-1: OSCCON: OSCILLATOR CONTROL REGISTER**

U-0	R/W-1	R/W-1	R/W-0	R-1	R-0	R-0	R/W-0
—	IRCF2	IRCF1	IRCF0	OSTS <sup>(1)</sup>	HTS	LTS	SCS
bit 7							bit 0

**Legend:**

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

bit 7 **Unimplemented:** Read as '0'

bit 6-4 **IRCF<2:0>:** Internal Oscillator Frequency Select bits

111 = 8 MHz

110 = 4 MHz (default)

101 = 2 MHz

100 = 1 MHz

011 = 500 kHz

010 = 250 kHz

001 = 125 kHz

000 = 31 kHz (LFINTOSC)

bit 3 **OSTS:** Oscillator Start-up Time-out Status bit<sup>(1)</sup>

1 = Device is running from the external clock defined by FOSC<2:0> of the Configuration Word register

0 = Device is running from the internal oscillator (HFINTOSC or LFINTOSC)

bit 2 **HTS:** HFINTOSC Status bit (High Frequency – 8 MHz to 125 kHz)

1 = HFINTOSC is stable

0 = HFINTOSC is not stable

bit 1 **LTS:** LFINTOSC Stable bit (Low Frequency – 31 kHz)

1 = LFINTOSC is stable

0 = LFINTOSC is not stable

bit 0 **SCS:** System Clock Select bit

1 = Internal oscillator is used for system clock

0 = Clock source defined by FOSC<2:0> of the Configuration Word register

**Note 1:** Bit resets to '0' with Two-Speed Start-up and LP, XT or HS selected as the Oscillator mode or Fail-Safe mode is enabled.



## 3.3 Clock Source Modes

Clock Source modes can be classified as external or internal.

- External Clock modes rely on external circuitry for the clock source. Examples are: Oscillator modules (EC mode), quartz crystal resonators or ceramic resonators (LP, XT and HS modes) and Resistor-Capacitor (RC) mode circuits.
- Internal clock sources are contained internally within the Oscillator module. The Oscillator module has two internal oscillators: the 8 MHz High-Frequency Internal Oscillator (HFINTOSC) and the 31 kHz Low-Frequency Internal Oscillator (LFINTOSC).

The system clock can be selected between external or internal clock sources via the System Clock Select (SCS) bit of the OSCCON register. See **Section 3.6 “Clock Switching”** for additional information.

## 3.4 External Clock Modes

### 3.4.1 OSCILLATOR START-UP TIMER (OST)

If the Oscillator module is configured for LP, XT or HS modes, the Oscillator Start-up Timer (OST) counts 1024 oscillations from OSC1. This occurs following a Power-on Reset (POR) and when the Power-up Timer (PWRT) has expired (if configured), or a wake-up from Sleep. During this time, the program counter does not increment and program execution is suspended. The OST ensures that the oscillator circuit, using a quartz crystal resonator or ceramic resonator, has started and is providing a stable system clock to the Oscillator module. When switching between clock sources, a delay is required to allow the new clock to stabilize. These oscillator delays are shown in Table 3-1.

In order to minimize latency between external oscillator start-up and code execution, the Two-Speed Clock Start-up mode can be selected (see **Section 3.7 “Two-Speed Clock Start-up Mode”**).

**TABLE 3-1: OSCILLATOR DELAY EXAMPLES**

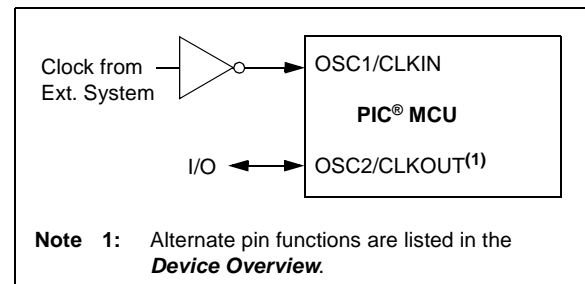
Switch From	Switch To	Frequency	Oscillator Delay
Sleep/POR	LFINTOSC HFINTOSC	31 kHz 125 kHz to 8 MHz	Oscillator Warm-Up Delay (TWARM)
Sleep/POR	EC, RC	DC – 20 MHz	2 instruction cycles
LFINTOSC (31 kHz)	EC, RC	DC – 20 MHz	1 cycle of each
Sleep/POR	LP, XT, HS	32 kHz to 20 MHz	1024 Clock Cycles (OST)
LFINTOSC (31 kHz)	HFINTOSC	125 kHz to 8 MHz	1 $\mu$ s (approx.)

### 3.4.2 EC MODE

The External Clock (EC) mode allows an externally generated logic level as the system clock source. When operating in this mode, an external clock source is connected to the OSC1 input and the OSC2 is available for general purpose I/O. Figure 3-2 shows the pin connections for EC mode.

The Oscillator Start-up Timer (OST) is disabled when EC mode is selected. Therefore, there is no delay in operation after a Power-on Reset (POR) or wake-up from Sleep. Because the PIC® MCU design is fully static, stopping the external clock input will have the effect of halting the device while leaving all data intact. Upon restarting the external clock, the device will resume operation as if no time had elapsed.

**FIGURE 3-2: EXTERNAL CLOCK (EC) MODE OPERATION**



# PIC12F683

## 3.4.3 LP, XT, HS MODES

The LP, XT and HS modes support the use of quartz crystal resonators or ceramic resonators connected to OSC1 and OSC2 (Figure 3-3). The mode selects a low, medium or high gain setting of the internal inverter-amplifier to support various resonator types and speed.

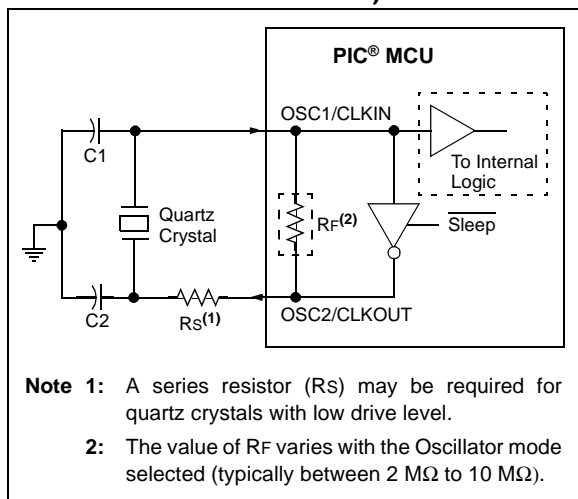
**LP** Oscillator mode selects the lowest gain setting of the internal inverter-amplifier. LP mode current consumption is the least of the three modes. This mode is designed to drive only 32.768 kHz tuning-fork type crystals (watch crystals).

**XT** Oscillator mode selects the intermediate gain setting of the internal inverter-amplifier. XT mode current consumption is the medium of the three modes. This mode is best suited to drive resonators with a medium drive level specification.

**HS** Oscillator mode selects the highest gain setting of the internal inverter-amplifier. HS mode current consumption is the highest of the three modes. This mode is best suited for resonators that require a high drive setting.

Figure 3-3 and Figure 3-4 show typical circuits for quartz crystal and ceramic resonators, respectively.

**FIGURE 3-3: QUARTZ CRYSTAL OPERATION (LP, XT OR HS MODE)**



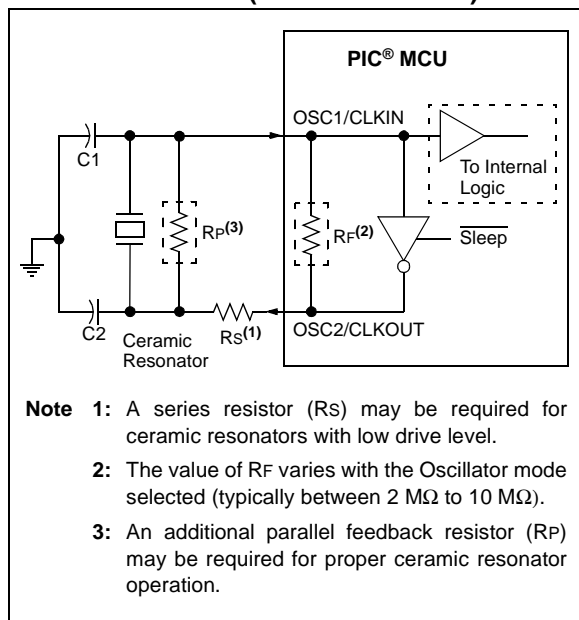
**Note 1:** Quartz crystal characteristics vary according to type, package and manufacturer. The user should consult the manufacturer data sheets for specifications and recommended application.

**2:** Always verify oscillator performance over the  $V_{DD}$  and temperature range that is expected for the application.

**3:** For oscillator design assistance, reference the following Microchip Applications Notes:

- AN826, "Crystal Oscillator Basics and Crystal Selection for *rPIC*<sup>®</sup> and *PIC*<sup>®</sup> Devices" (DS00826)
- AN849, "Basic *PIC*<sup>®</sup> Oscillator Design" (DS00849)
- AN943, "Practical *PIC*<sup>®</sup> Oscillator Analysis and Design" (DS00943)
- AN949, "Making Your Oscillator Work" (DS00949)

**FIGURE 3-4: CERAMIC RESONATOR OPERATION (XT OR HS MODE)**

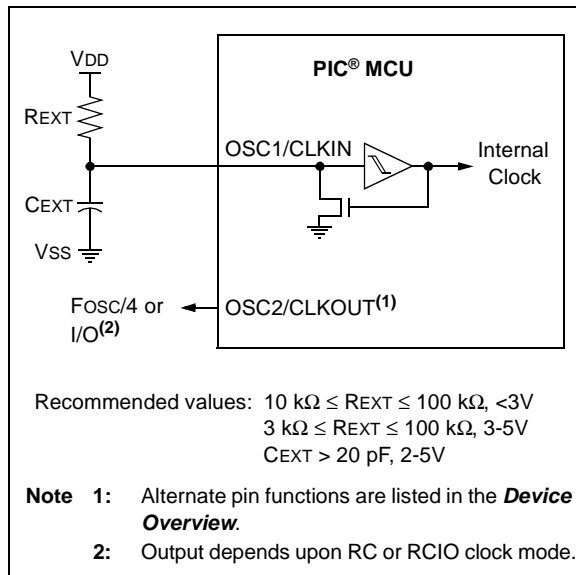


## 3.4.4 EXTERNAL RC MODES

The external Resistor-Capacitor (RC) modes support the use of an external RC circuit. This allows the designer maximum flexibility in frequency choice while keeping costs to a minimum when clock accuracy is not required. There are two modes: RC and RCIO.

In RC mode, the RC circuit connects to OSC1. OSC2/CLKOUT outputs the RC oscillator frequency divided by 4. This signal may be used to provide a clock for external circuitry, synchronization, calibration, test or other application requirements. Figure 3-5 shows the external RC mode connections.

**FIGURE 3-5: EXTERNAL RC MODES**



In RCIO mode, the RC circuit is connected to OSC1. OSC2 becomes an additional general purpose I/O pin.

The RC oscillator frequency is a function of the supply voltage, the resistor ( $R_{EXT}$ ) and capacitor ( $C_{EXT}$ ) values and the operating temperature. Other factors affecting the oscillator frequency are:

- threshold voltage variation
- component tolerances
- packaging variations in capacitance

The user also needs to take into account variation due to tolerance of external RC components used.

## 3.5 Internal Clock Modes

The Oscillator module has two independent, internal oscillators that can be configured or selected as the system clock source.

1. The **HFINTOSC** (High-Frequency Internal Oscillator) is factory calibrated and operates at 8 MHz. The frequency of the HFINTOSC can be user-adjusted via software using the OSCTUNE register (Register 3-2).
2. The **LFINTOSC** (Low-Frequency Internal Oscillator) is uncalibrated and operates at 31 kHz.

The system clock speed can be selected via software using the Internal Oscillator Frequency Select bits  $IRCF<2:0>$  of the OSCCON register.

The system clock can be selected between external or internal clock sources via the System Clock Selection (SCS) bit of the OSCCON register. See **Section 3.6 “Clock Switching”** for more information.

### 3.5.1 INTOSC AND INTOSCIO MODES

The INTOSC and INTOSCIO modes configure the internal oscillators as the system clock source when the device is programmed using the oscillator selection or the  $FOSC<2:0>$  bits in the Configuration Word register (CONFIG). See **Section 12.0 “Special Features of the CPU”** for more information.

In **INTOSC** mode, OSC1/CLKIN is available for general purpose I/O. OSC2/CLKOUT outputs the selected internal oscillator frequency divided by 4. The CLKOUT signal may be used to provide a clock for external circuitry, synchronization, calibration, test or other application requirements.

In **INTOSCIO** mode, OSC1/CLKIN and OSC2/CLKOUT are available for general purpose I/O.

### 3.5.2 HFINTOSC

The High-Frequency Internal Oscillator (HFINTOSC) is a factory calibrated 8 MHz internal clock source. The frequency of the HFINTOSC can be altered via software using the OSCTUNE register (Register 3-2).

The output of the HFINTOSC connects to a postscaler and multiplexer (see Figure 3-1). One of seven frequencies can be selected via software using the  $IRCF<2:0>$  bits of the OSCCON register. See **Section 3.5.4 “Frequency Select Bits (IRCF)”** for more information.

The HFINTOSC is enabled by selecting any frequency between 8 MHz and 125 kHz by setting the  $IRCF<2:0>$  bits of the OSCCON register  $\neq 000$ . Then, set the System Clock Source (SCS) bit of the OSCCON register to ‘1’ or enable Two-Speed Start-up by setting the IESO bit in the Configuration Word register (CONFIG) to ‘1’.

The HF Internal Oscillator (HTS) bit of the OSCCON register indicates whether the HFINTOSC is stable or not.

# PIC12F683

## 3.5.2.1 OSCTUNE Register

The HFINTOSC is factory calibrated but can be adjusted in software by writing to the OSCTUNE register (Register 3-2).

The default value of the OSCTUNE register is '0'. The value is a 5-bit two's complement number.

When the OSCTUNE register is modified, the HFINTOSC frequency will begin shifting to the new frequency. Code execution continues during this shift. There is no indication that the shift has occurred.

OSCTUNE does not affect the LFINTOSC frequency. Operation of features that depend on the LFINTOSC clock source frequency, such as the Power-up Timer (PWRT), Watchdog Timer (WDT), Fail-Safe Clock Monitor (FSCM) and peripherals, are *not* affected by the change in frequency.

### REGISTER 3-2: OSCTUNE: OSCILLATOR TUNING REGISTER

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	TUN4	TUN3	TUN2	TUN1	TUN0
bit 7							
							bit 0

#### Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

bit 7-5 **Unimplemented:** Read as '0'

bit 4-0 **TUN<4:0>:** Frequency Tuning bits

01111 = Maximum frequency

01110 =

•

•

•

00001 =

00000 = Oscillator module is running at the calibrated frequency.

11111 =

•

•

•

10000 = Minimum frequency

## 3.5.3 LFINTOSC

The Low-Frequency Internal Oscillator (LFINTOSC) is an uncalibrated 31 kHz internal clock source.

The output of the LFINTOSC connects to a postscaler and multiplexer (see Figure 3-1). Select 31 kHz, via software, using the IRCF<2:0> bits of the OSCCON register. See **Section 3.5.4 “Frequency Select Bits (IRCF)”** for more information. The LFINTOSC is also the frequency for the Power-up Timer (PWRT), Watchdog Timer (WDT) and Fail-Safe Clock Monitor (FSCM).

The LFINTOSC is enabled by selecting 31 kHz (IRCF<2:0> bits of the OSCCON register = 000) as the system clock source (SCS bit of the OSCCON register = 1), or when any of the following are enabled:

- Two-Speed Start-up IESO bit of the Configuration Word register = 1 and IRCF<2:0> bits of the OSCCON register = 000
- Power-up Timer (PWRT)
- Watchdog Timer (WDT)
- Fail-Safe Clock Monitor (FSCM)

The LF Internal Oscillator (LTS) bit of the OSCCON register indicates whether the LFINTOSC is stable or not.

## 3.5.4 FREQUENCY SELECT BITS (IRCF)

The output of the 8 MHz HFINTOSC and 31 kHz LFINTOSC connects to a postscaler and multiplexer (see Figure 3-1). The Internal Oscillator Frequency Select bits IRCF<2:0> of the OSCCON register select the frequency output of the internal oscillators. One of eight frequencies can be selected via software:

- 8 MHz
- 4 MHz (Default after Reset)
- 2 MHz
- 1 MHz
- 500 kHz
- 250 kHz
- 125 kHz
- 31 kHz (LFINTOSC)

**Note:** Following any Reset, the IRCF<2:0> bits of the OSCCON register are set to '110' and the frequency selection is set to 4 MHz. The user can modify the IRCF bits to select a different frequency.

## 3.5.5 HF AND LF INTOSC CLOCK SWITCH TIMING

When switching between the LFINTOSC and the HFINTOSC, the new oscillator may already be shut down to save power (see Figure 3-6). If this is the case, there is a delay after the IRCF<2:0> bits of the OSCCON register are modified before the frequency selection takes place. The LTS and HTS bits of the OSCCON register will reflect the current active status of the LFINTOSC and HFINTOSC oscillators. The timing of a frequency selection is as follows:

1. IRCF<2:0> bits of the OSCCON register are modified.
2. If the new clock is shut down, a clock start-up delay is started.
3. Clock switch circuitry waits for a falling edge of the current clock.
4. CLKOUT is held low and the clock switch circuitry waits for a rising edge in the new clock.
5. CLKOUT is now connected with the new clock. LTS and HTS bits of the OSCCON register are updated as required.
6. Clock switch is complete.

See Figure 3-1 for more details.

If the internal oscillator speed selected is between 8 MHz and 125 kHz, there is no start-up delay before the new frequency is selected. This is because the old and new frequencies are derived from the HFINTOSC via the postscaler and multiplexer.

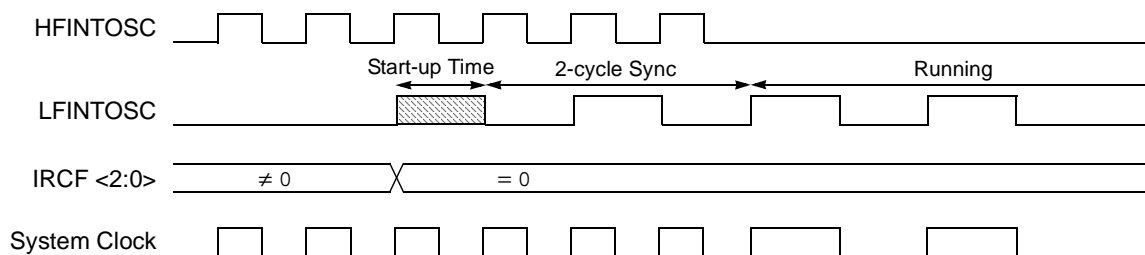
Start-up delay specifications are located in the **Electrical Specifications Chapter of this data sheet, under AC Specifications (Oscillator Module)**.

# PIC12F683

**FIGURE 3-6: INTERNAL OSCILLATOR SWITCH TIMING**

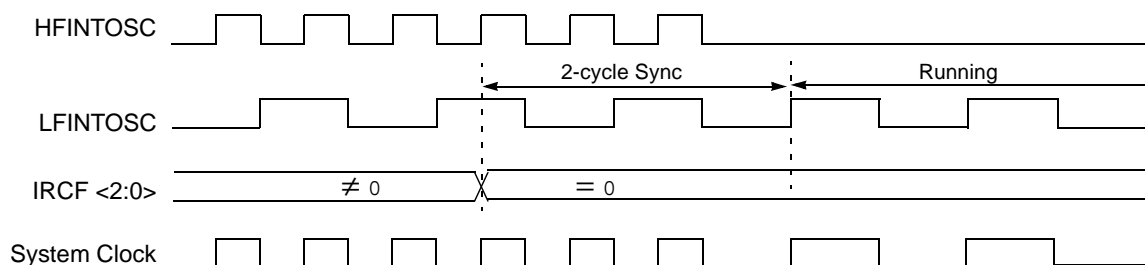
**HF → LF<sup>(1)</sup>**

**HFINTOSC → LFINTOSC (FSCM and WDT disabled)**

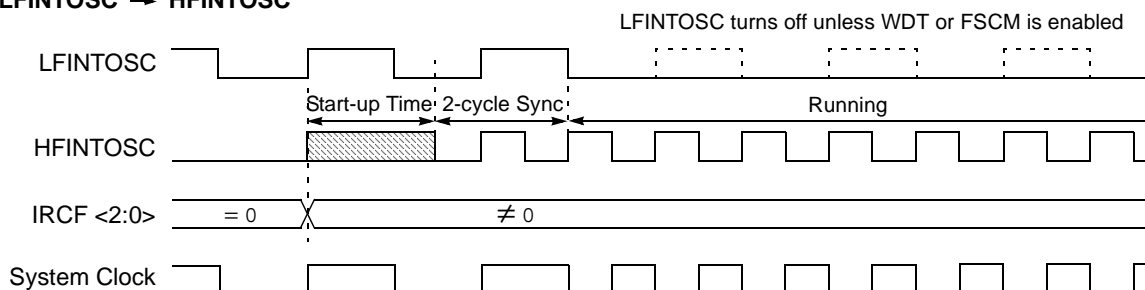


**Note 1:** When going from LF to HF.

**HFINTOSC → LFINTOSC (Either FSCM or WDT enabled)**



**LFINTOSC → HFINTOSC**



## 3.6 Clock Switching

The system clock source can be switched between external and internal clock sources via software using the System Clock Select (SCS) bit of the OSCCON register.

### 3.6.1 SYSTEM CLOCK SELECT (SCS) BIT

The System Clock Select (SCS) bit of the OSCCON register selects the system clock source that is used for the CPU and peripherals.

- When the SCS bit of the OSCCON register = 0, the system clock source is determined by configuration of the FOSC<2:0> bits in the Configuration Word register (CONFIG).
- When the SCS bit of the OSCCON register = 1, the system clock source is chosen by the internal oscillator frequency selected by the IRCF<2:0> bits of the OSCCON register. After a Reset, the SCS bit of the OSCCON register is always cleared.

**Note:** Any automatic clock switch, which may occur from Two-Speed Start-up or Fail-Safe Clock Monitor, does not update the SCS bit of the OSCCON register. The user can monitor the OSTS bit of the OSCCON register to determine the current system clock source.

### 3.6.2 OSCILLATOR START-UP TIME-OUT STATUS (OSTS) BIT

The Oscillator Start-up Time-out Status (OSTS) bit of the OSCCON register indicates whether the system clock is running from the external clock source, as defined by the FOSC<2:0> bits in the Configuration Word register (CONFIG), or from the internal clock source. In particular, OSTS indicates that the Oscillator Start-up Timer (OST) has timed out for LP, XT or HS modes.

## 3.7 Two-Speed Clock Start-up Mode

Two-Speed Start-up mode provides additional power savings by minimizing the latency between external oscillator start-up and code execution. In applications that make heavy use of the Sleep mode, Two-Speed Start-up will remove the external oscillator start-up time from the time spent awake and can reduce the overall power consumption of the device.

This mode allows the application to wake-up from Sleep, perform a few instructions using the INTOSC as the clock source and go back to Sleep without waiting for the primary oscillator to become stable.

**Note:** Executing a SLEEP instruction will abort the oscillator start-up time and will cause the OSTS bit of the OSCCON register to remain clear.

When the Oscillator module is configured for LP, XT or HS modes, the Oscillator Start-up Timer (OST) is enabled (see **Section 3.4.1 “Oscillator Start-up Timer (OST)”**). The OST will suspend program execution until 1024 oscillations are counted. Two-Speed Start-up mode minimizes the delay in code execution by operating from the internal oscillator as the OST is counting. When the OST count reaches 1024 and the OSTS bit of the OSCCON register is set, program execution switches to the external oscillator.

### 3.7.1 TWO-SPEED START-UP MODE CONFIGURATION

Two-Speed Start-up mode is configured by the following settings:

- IESO (of the Configuration Word register) = 1; Internal/External Switchover bit (Two-Speed Start-up mode enabled).
- SCS (of the OSCCON register) = 0.
- FOSC<2:0> bits in the Configuration Word register (CONFIG) configured for LP, XT or HS mode.

Two-Speed Start-up mode is entered after:

- Power-on Reset (POR) and, if enabled, after Power-up Timer (PWRT) has expired, or
- Wake-up from Sleep.

If the external clock oscillator is configured to be anything other than LP, XT or HS mode, then Two-Speed Start-up is disabled. This is because the external clock oscillator does not require any stabilization time after POR or an exit from Sleep.

### 3.7.2 TWO-SPEED START-UP SEQUENCE

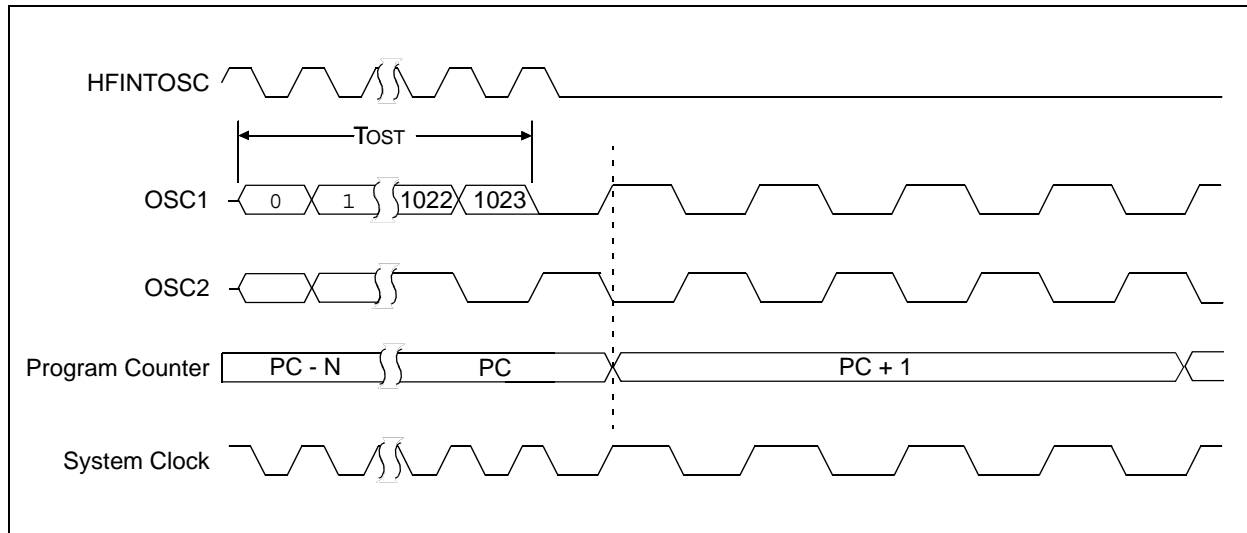
1. Wake-up from Power-on Reset or Sleep.
2. Instructions begin execution by the internal oscillator at the frequency set in the IRCF<2:0> bits of the OSCCON register.
3. OST enabled to count 1024 clock cycles.
4. OST timed out, wait for falling edge of the internal oscillator.
5. OSTS is set.
6. System clock held low until the next falling edge of new clock (LP, XT or HS mode).
7. System clock is switched to external clock source.

# PIC12F683

## 3.7.3 CHECKING TWO-SPEED CLOCK STATUS

Checking the state of the OSTS bit of the OSCCON register will confirm if the microcontroller is running from the external clock source, as defined by the FOSC<2:0> bits in the Configuration Word register (CONFIG), or the internal oscillator.

**FIGURE 3-7: TWO-SPEED START-UP**

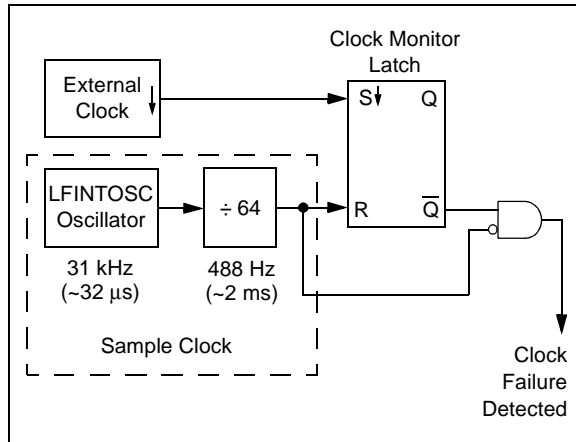




## 3.8 Fail-Safe Clock Monitor

The Fail-Safe Clock Monitor (FSCM) allows the device to continue operating should the external oscillator fail. The FSCM can detect oscillator failure any time after the Oscillator Start-up Timer (OST) has expired. The FSCM is enabled by setting the FCMEN bit in the Configuration Word register (CONFIG). The FSCM is applicable to all external oscillator modes (LP, XT, HS, EC, RC and RCIO).

**FIGURE 3-8: FSCM BLOCK DIAGRAM**



### 3.8.1 FAIL-SAFE DETECTION

The FSCM module detects a failed oscillator by comparing the external oscillator to the FSCM sample clock. The sample clock is generated by dividing the LFINTOSC by 64. See Figure 3-8. Inside the fail detector block is a latch. The external clock sets the latch on each falling edge of the external clock. The sample clock clears the latch on each rising edge of the sample clock. A failure is detected when an entire half-cycle of the sample clock elapses before the primary clock goes low.

### 3.8.2 FAIL-SAFE OPERATION

When the external clock fails, the FSCM switches the device clock to an internal clock source and sets the bit flag OSFIF of the PIR1 register. Setting this flag will generate an interrupt if the OSFIE bit of the PIE1 register is also set. The device firmware can then take steps to mitigate the problems that may arise from a failed clock. The system clock will continue to be sourced from the internal clock source until the device firmware successfully restarts the external oscillator and switches back to external operation.

The internal clock source chosen by the FSCM is determined by the IRCF<2:0> bits of the OSCCON register. This allows the internal oscillator to be configured before a failure occurs.

### 3.8.3 FAIL-SAFE CONDITION CLEARING

The Fail-Safe condition is cleared after a Reset, executing a `SLEEP` instruction or toggling the SCS bit of the OSCCON register. When the SCS bit is toggled, the OST is restarted. While the OST is running, the device continues to operate from the INTOSC selected in OSCCON. When the OST times out, the Fail-Safe condition is cleared and the device will be operating from the external clock source. The Fail-Safe condition must be cleared before the OSFIF flag can be cleared.

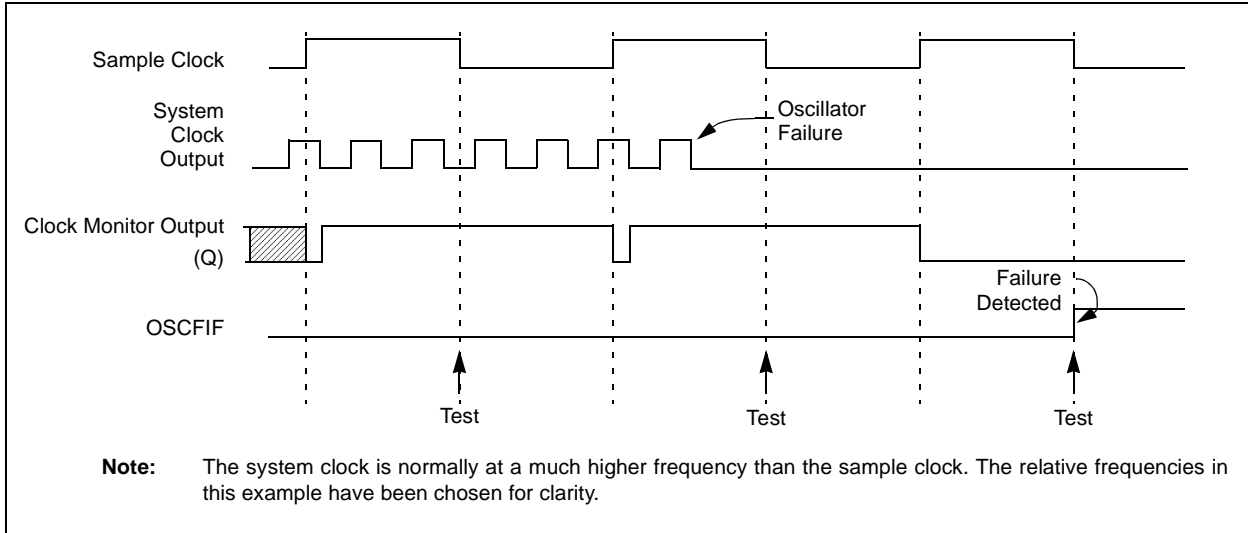
### 3.8.4 RESET OR WAKE-UP FROM SLEEP

The FSCM is designed to detect an oscillator failure after the Oscillator Start-up Timer (OST) has expired. The OST is used after waking up from Sleep and after any type of Reset. The OST is not used with the EC or RC Clock modes so that the FSCM will be active as soon as the Reset or wake-up has completed. When the FSCM is enabled, the Two-Speed Start-up is also enabled. Therefore, the device will always be executing code while the OST is operating.

**Note:** Due to the wide range of oscillator start-up times, the Fail-Safe circuit is not active during oscillator start-up (i.e., after exiting Reset or Sleep). After an appropriate amount of time, the user should check the OSTS bit of the OSCCON register to verify the oscillator start-up and that the system clock switchover has successfully completed.

# PIC12F683

**FIGURE 3-9: FSCM TIMING DIAGRAM**



**TABLE 3-2: SUMMARY OF REGISTERS ASSOCIATED WITH CLOCK SOURCES**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets <sup>(1)</sup>
CONFIG <sup>(2)</sup>	CPD	CP	MCLRE	PWRTE	WDTE	FOSC2	FOSC1	FOSC0	—	—
INTCON	GIE	PEIE	T0IE	INTE	GPIE	T0IF	INTF	GPIF	0000 0000	0000 000x
OSCCON	—	IRCF2	IRCF1	IRCF0	OSTS	HTS	LTS	SCS	-110 x000	-110 x000
OSCTUNE	—	—	—	TUN4	TUN3	TUN2	TUN1	TUN0	---0 0000	---u uuuu
PIE1	EEIE	ADIE	CCP1IE	—	CMIE	OSFIE	TMR2IE	TMR1IE	000- 0000	000- 0000
PIR1	EEIF	ADIF	CCP1IF	—	CMIF	OSFIF	TMR2IF	TMR1IF	000- 0000	000- 0000

**Legend:** x = unknown, u = unchanged, - = unimplemented locations read as '0'. Shaded cells are not used by oscillators.

**Note 1:** Other (non Power-up) Resets include MCLR Reset and Watchdog Timer Reset during normal operation.

**2:** See Configuration Word register (Register 12-1) for operation of all register bits.

## 4.0 GPIO PORT

There are as many as six general purpose I/O pins available. Depending on which peripherals are enabled, some or all of the pins may not be available as general purpose I/O. In general, when a peripheral is enabled, the associated pin may not be used as a general purpose I/O pin.

### 4.1 GPIO and the TRISIO Registers

GPIO is a 6-bit wide, bidirectional port. The corresponding data direction register is TRISIO. Setting a TRISIO bit (= 1) will make the corresponding GPIO pin an input (i.e., put the corresponding output driver in a High-Impedance mode). Clearing a TRISIO bit (= 0) will make the corresponding GPIO pin an output (i.e., put the contents of the output latch on the selected pin). An exception is GP3, which is input only and its TRISIO bit will always read as '1'. Example 4-1 shows how to initialize GPIO.

Reading the GPIO register reads the status of the pins, whereas writing to it will write to the PORT latch. All write operations are read-modify-write operations.

Therefore, a write to a port implies that the port pins are read, this value is modified and then written to the PORT data latch. GP3 reads '0' when MCLRE = 1.

The TRISIO register controls the direction of the GPIO pins, even when they are being used as analog inputs. The user must ensure the bits in the TRISIO register are maintained set when using them as analog inputs. I/O pins configured as analog input always read '0'.

**Note:** The ANSEL and CMCON0 registers must be initialized to configure an analog channel as a digital input. Pins configured as analog inputs will read '0'.

#### EXAMPLE 4-1: INITIALIZING GPIO

```
BANKSEL GPIO      ;
CLRF  GPIO        ;Init GPIO
MOVLW 07h         ;Set GP<2:0> to
MOVWF CMCON0      ;digital I/O
BANKSEL ANSEL     ;
CLRF  ANSEL       ;digital I/O
MOVLW 0Ch         ;Set GP<3:2> as inputs
MOVWF TRISIO      ;and set GP<5:4,1:0>
                  ;as outputs
```

#### REGISTER 4-1: GPIO: GENERAL PURPOSE I/O REGISTER

U-0	U-0	R/W-x	R/W-0	R-x	R/W-0	R/W-0	R/W-0
—	—	GP5	GP4	GP3	GP2	GP1	GP0
bit 7							bit 0

##### Legend:

R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared
		x = Bit is unknown

bit 7-6      **Unimplemented:** Read as '0'

bit 5-0      **GP<5:0>:** GPIO I/O Pin bit

             1 = Port pin is > V<sub>IH</sub>

             0 = Port pin is < V<sub>IL</sub>

# PIC12F683

## REGISTER 4-2: TRISIO GPIO TRI-STATE REGISTER

U-0	U-0	R/W-1	R/W-1	R-1	R/W-1	R/W-1	R/W-1
—	—	TRISIO5 <sup>(2,3)</sup>	TRISIO4 <sup>(2)</sup>	TRISIO3 <sup>(1)</sup>	TRISIO2	TRISIO1	TRISIO0
bit 7		bit 0					

### Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

bit 7-6 **Unimplemented:** Read as '0'

bit 5:4 **TRISIO<5:4>:** GPIO Tri-State Control bit  
1 = GPIO pin configured as an input (tri-stated)  
0 = GPIO pin configured as an output

bit 3 **TRISIO<3>:** GPIO Tri-State Control bit  
Input only

bit 2:0 **TRISIO<2:0>:** GPIO Tri-State Control bit  
1 = GPIO pin configured as an input (tri-stated)  
0 = GPIO pin configured as an output

- Note** 1: TRISIO<3> always reads '1'.  
2: TRISIO<5:4> always reads '1' in XT, HS and LP OSC modes.  
3: TRISIO<5> always reads '1' in RC and RCIO and EC modes.

## 4.2 Additional Pin Functions

Every GPIO pin on the PIC12F683 has an interrupt-on-change option and a weak pull-up option. GP0 has an Ultra Low-Power Wake-up option. The next three sections describe these functions.

### 4.2.1 ANSEL REGISTER

The ANSEL register is used to configure the Input mode of an I/O pin to analog. Setting the appropriate ANSEL bit high will cause all digital reads on the pin to be read as '0' and allow analog functions on the pin to operate correctly.

The state of the ANSEL bits has no affect on digital output functions. A pin with TRIS clear and ANSEL set will still operate as a digital output, but the Input mode will be analog. This can cause unexpected behavior when executing read-modify-write instructions on the affected port.

### 4.2.2 WEAK PULL-UPS

Each of the GPIO pins, except GP3, has an individually configurable internal weak pull-up. Control bits WPUx enable or disable each pull-up. Refer to Register 4-4. Each weak pull-up is automatically turned off when the port pin is configured as an output. The pull-ups are disabled on a Power-on Reset by the  $\overline{\text{GPPU}}$  bit of the OPTION register). A weak pull-up is automatically enabled for GP3 when configured as MCLR and disabled when GP3 is an I/O. There is no software control of the MCLR pull-up.

### 4.2.3 INTERRUPT-ON-CHANGE

Each of the GPIO pins is individually configurable as an interrupt-on-change pin. Control bits IOCx enable or disable the interrupt function for each pin. Refer to Register 4-5. The interrupt-on-change is disabled on a Power-on Reset.

For enabled interrupt-on-change pins, the values are compared with the old value latched on the last read of GPIO. The 'mismatch' outputs of the last read are OR'd together to set the GPIO Change Interrupt Flag bit (GPIF) in the INTCON register (Register 2-3).

This interrupt can wake the device from Sleep. The user, in the Interrupt Service Routine, clears the interrupt by:

- Any read or write of GPIO. This will end the mismatch condition, then,
- Clear the flag bit GPIF.

A mismatch condition will continue to set flag bit GPIF. Reading GPIO will end the mismatch condition and allow flag bit GPIF to be cleared. The latch holding the last read value is not affected by a MCLR nor Brown-out Reset. After these resets, the GPIF flag will continue to be set if a mismatch is present.

**Note:** If a change on the I/O pin should occur when any GPIO operation is being executed, then the GPIF interrupt flag may not get set.

## REGISTER 4-3: ANSEL: ANALOG SELECT REGISTER

U-0	R/W-0	R/W-0	R/W-0	R/W-1	R/W-1	R/W-1	R/W-1
—	ADCS2	ADCS1	ADCS0	ANS3	ANS2	ANS1	ANS0
bit 7							bit 0

### Legend:

R = Readable bit                      W = Writable bit                      U = Unimplemented bit, read as '0'  
 -n = Value at POR                      '1' = Bit is set                      '0' = Bit is cleared                      x = Bit is unknown

bit 7                      **Unimplemented:** Read as '0'

bit 6-4                      **ADCS<2:0>:** A/D Conversion Clock Select bits

000 = FOSC/2  
 001 = FOSC/8  
 010 = FOSC/32  
 x11 = FRC (clock derived from a dedicated internal oscillator = 500 kHz max)  
 100 = FOSC/4  
 101 = FOSC/16  
 110 = FOSC/64

bit 3-0                      **ANS<3:0>:** Analog Select bits

Analog select between analog or digital function on pins AN<3:0>, respectively.

1 = Analog input. Pin is assigned as analog input<sup>(1)</sup>.

0 = Digital I/O. Pin is assigned to port or special function.

**Note 1:** Setting a pin to an analog input automatically disables the digital input circuitry, weak pull-ups and interrupt-on-change, if available. The corresponding TRIS bit must be set to Input mode in order to allow external control of the voltage on the pin.

# PIC12F683

## REGISTER 4-4: WPU: WEAK PULL-UP REGISTER

U-0	U-0	R/W-1	R/W-1	U-0	R/W-1	R/W-1	R/W-1
—	—	WPU5	WPU4	—	WPU2	WPU1	WPU0
bit 7							bit 0

### Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

bit 7-6 **Unimplemented:** Read as '0'

bit 5-4 **WPU<5:4>:** Weak Pull-up Control bits

1 = Pull-up enabled

0 = Pull-up disabled

bit 3 **Unimplemented:** Read as '0'

bit 2-0 **WPU<2:0>:** Weak Pull-up Control bits

1 = Pull-up enabled

0 = Pull-up disabled

**Note 1:** Global GPPU must be enabled for individual pull-ups to be enabled.

**2:** The weak pull-up device is automatically disabled if the pin is in Output mode (TRISIO = 0).

**3:** The GP3 pull-up is enabled when configured as  $\overline{\text{MCLR}}$  and disabled as an I/O in the Configuration Word.

**4:** WPU<5:4> always reads '1' in XT, HS and LP OSC modes.

## REGISTER 4-5: IOC: INTERRUPT-ON-CHANGE GPIO REGISTER

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	IOC5	IOC4	IOC3	IOC2	IOC1	IOC0
bit 7							bit 0

### Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

bit 7-6 **Unimplemented:** Read as '0'

bit 5-0 **IOC<5:0>:** Interrupt-on-change GPIO Control bits

1 = Interrupt-on-change enabled

0 = Interrupt-on-change disabled

**Note 1:** Global Interrupt Enable (GIE) must be enabled for individual interrupts to be recognized.

**2:** IOC<5:4> always reads '0' in XT, HS and LP OSC modes.

## 4.2.4 ULTRA LOW-POWER WAKE-UP

The Ultra Low-Power Wake-up (ULPWU) on GP0 allows a slow falling voltage to generate an interrupt-on-change on GP0 without excess current consumption. The mode is selected by setting the ULPWUE bit of the PCON register. This enables a small current sink which can be used to discharge a capacitor on GP0.

To use this feature, the GP0 pin is configured to output '1' to charge the capacitor, interrupt-on-change for GP0 is enabled and GP0 is configured as an input. The ULPWUE bit is set to begin the discharge and a SLEEP instruction is performed. When the voltage on GP0 drops below  $V_{IL}$ , an interrupt will be generated which will cause the device to wake-up. Depending on the state of the GIE bit of the INTCON register, the device will either jump to the interrupt vector (0004h) or execute the next instruction when the interrupt event occurs. See **Section 4.2.3 "Interrupt-on-Change"** and **Section 12.4.3 "GPIO Interrupt"** for more information.

This feature provides a low-power technique for periodically waking up the device from Sleep. The time-out is dependent on the discharge time of the RC circuit on GP0. See Example 4-2 for initializing the Ultra Low-Power Wake-up module.

The series resistor provides overcurrent protection for the GP0 pin and can allow for software calibration of the time-out (see Figure 4-1). A timer can be used to measure the charge time and discharge time of the capacitor. The charge time can then be adjusted to provide the desired interrupt delay. This technique will compensate for the affects of temperature, voltage and component accuracy. The Ultra Low-Power Wake-up peripheral can also be configured as a simple Programmable Low-Voltage Detect or temperature sensor.

**Note:** For more information, refer to the Application Note AN879, "Using the Microchip Ultra Low-Power Wake-up Module" (DS00879).

### EXAMPLE 4-2: ULTRA LOW-POWER WAKE-UP INITIALIZATION

```
BANKSEL CMCON0      ;
MOVLW  H'7'          ;Turn off
MOVWF  CMCON0        ;comparators
BANKSEL ANSEL        ;
BCF     ANSEL,0       ;RA0 to digital I/O
BCF     TRISA,0       ;Output high to
BANKSEL PORTA        ;
BSF     PORTA,0       ;charge capacitor
CALL    CapDelay      ;
BANKSEL PCON          ;
BSF     PCON,ULPWUE   ;Enable ULP Wake-up
BSF     IOCA,0        ;Select RA0 IOC
BSF     TRISA,0       ;RA0 to input
MOVLW  B'10001000'   ;Enable interrupt
MOVWF  INTCON         ; and clear flag
SLEEP                               ;Wait for IOC
NOP                               ;
```

# PIC12F683

#### 4.2.5 PIN DESCRIPTIONS AND DIAGRAMS

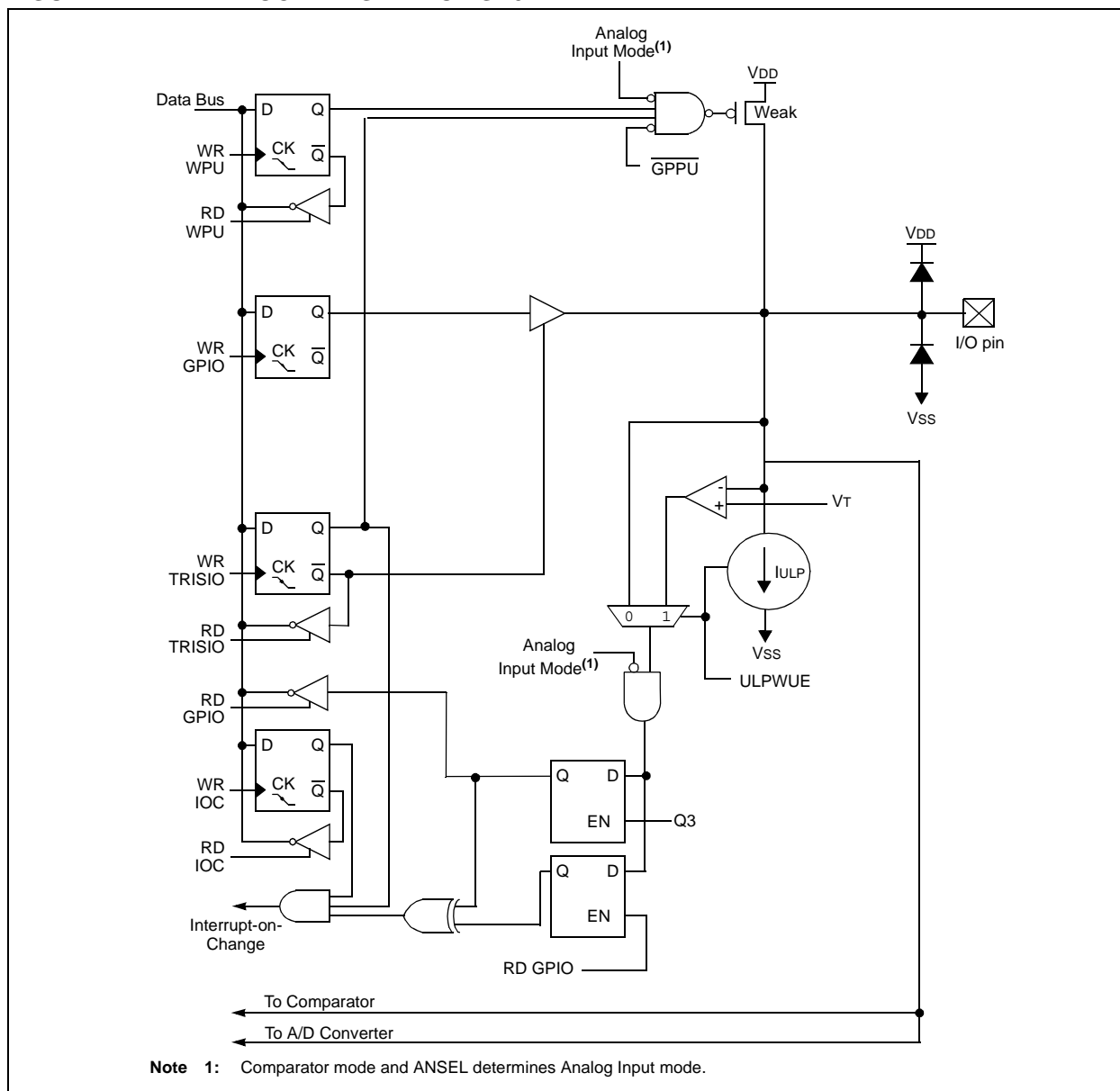
Each GPIO pin is multiplexed with other functions. The pins and their combined functions are briefly described here. For specific information about individual functions such as the comparator or the ADC, refer to the appropriate section in this data sheet.

#### 4.2.5.1 GP0/AN0/CIN+/ICSPDAT/ULPWU

Figure 4-1 shows the diagram for this pin. The GP0 pin is configurable to function as one of the following:

- a general purpose I/O
- an analog input for the ADC
- an analog input to the comparator
- In-Circuit Serial Programming™ data
- an analog input to the Ultra Low-Power Wake-up

**FIGURE 4-1: BLOCK DIAGRAM OF GP0**







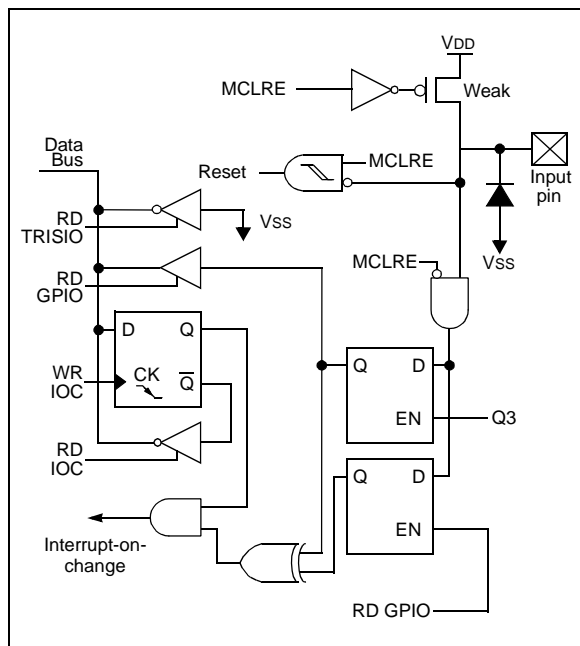
# PIC12F683

## 4.2.5.4 GP3/MCLR/VPP

Figure 4-4 shows the diagram for this pin. The GP3 pin is configurable to function as one of the following:

- a general purpose input
- as Master Clear Reset with weak pull-up

**FIGURE 4-4: BLOCK DIAGRAM OF GP3**

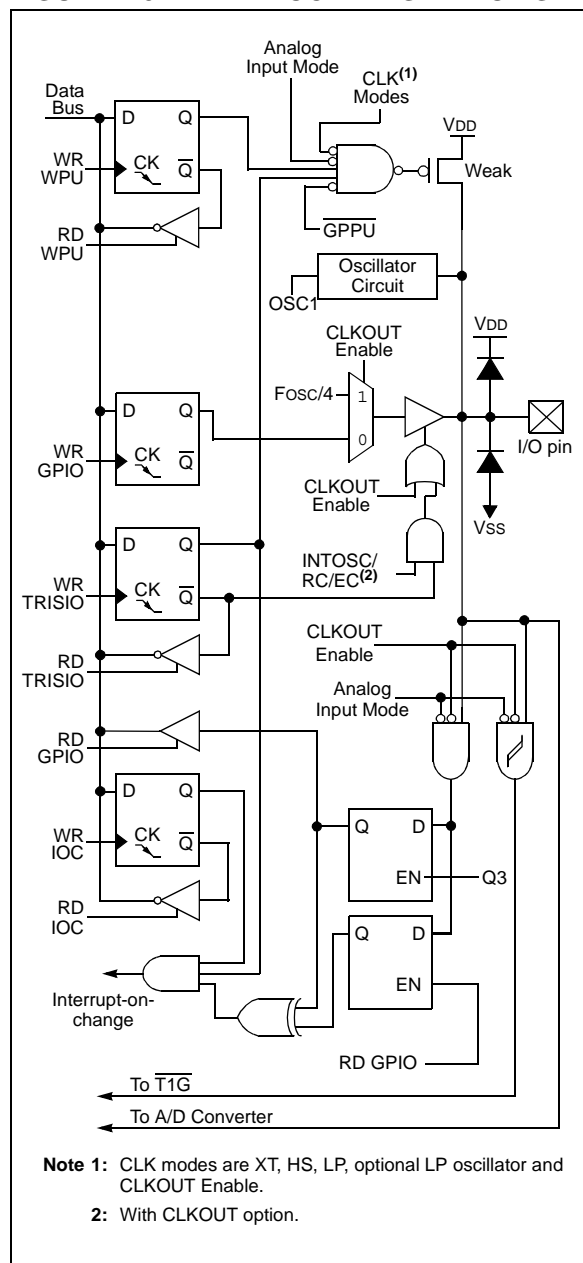


## 4.2.5.5 GP4/AN3/T1G/OSC2/CLKOUT

Figure 4-5 shows the diagram for this pin. The GP4 pin is configurable to function as one of the following:

- a general purpose I/O
- an analog input for the ADC
- a Timer1 gate input
- a crystal/resonator connection
- a clock output

**FIGURE 4-5: BLOCK DIAGRAM OF GP4**

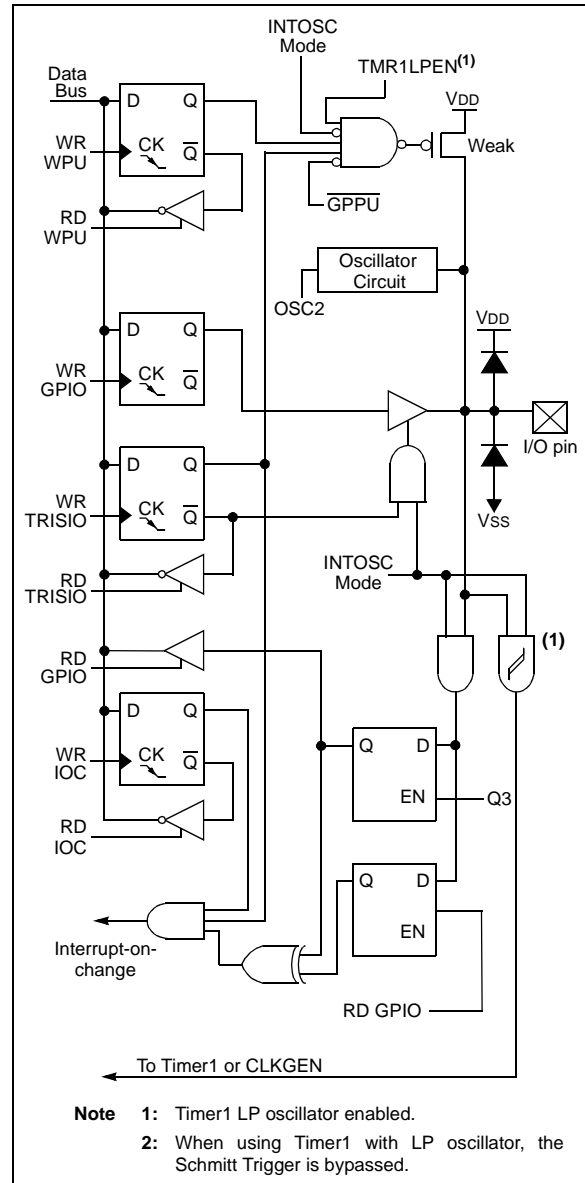


## 4.2.5.6 GP5/T1CKI/OSC1/CLKIN

Figure 4-6 shows the diagram for this pin. The GP5 pin is configurable to function as one of the following:

- a general purpose I/O
- a Timer1 clock input
- a crystal/resonator connection
- a clock input

**FIGURE 4-6: BLOCK DIAGRAM OF GP5**



**TABLE 4-1: SUMMARY OF REGISTERS ASSOCIATED WITH GPIO**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
ANSEL	—	ADCS2	ADCS1	ADCS0	ANS3	ANS2	ANS1	ANS0	-000 1111	-000 1111
CCP1CON	—	—	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	--00 0000
CMCON0	—	COU	—	CINV	CIS	CM2	CM1	CM0	-0-0 0000	-0-0 0000
PCON	—	—	ULPWUE	SBOREN	—	—	POR	BOR	--01 --qq	--0u --uu
INTCON	GIE	PEIE	T0IE	INTE	GPIOE	T0IF	INTF	GPIOF	0000 0000	0000 000x
IOC	—	—	IOC5	IOC4	IOC3	IOC2	IOC1	IOC0	--00 0000	--00 0000
OPTION_REG	GPPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
GPIO	—	—	GP5	GP4	GP3	GP2	GP1	GP0	--xx xxxx	--x0 x000
T1CON	T1GINV	TMR1GE	T1CKPS1	T1CKPS0	T1OSCEN	T1SYN	TMR1CS	TMR1ON	0000 0000	0000 0000
TRISIO	—	—	TRISIO5	TRISIO4	TRISIO3	TRISIO2	TRISIO1	TRISIO0	--11 1111	--11 1111
WPU	—	—	WPU5	WPU4	—	WPU2	WPU1	WPU0	--11 -111	--11 -111

**Legend:** x = unknown, u = unchanged, — = unimplemented locations read as '0'. Shaded cells are not used by GPIO.

# PIC12F683

---

NOTES:

## 5.0 TIMER0 MODULE

The Timer0 module is an 8-bit timer/counter with the following features:

- 8-bit timer/counter register (TMR0)
- 8-bit prescaler (shared with Watchdog Timer)
- Programmable internal or external clock source
- Programmable external clock edge selection
- Interrupt on overflow

Figure 5-1 is a block diagram of the Timer0 module.

## 5.1 Timer0 Operation

When used as a timer, the Timer0 module can be used as either an 8-bit timer or an 8-bit counter.

### 5.1.1 8-BIT TIMER MODE

When used as a timer, the Timer0 module will increment every instruction cycle (without prescaler). Timer mode is selected by clearing the T0CS bit of the OPTION register to '0'.

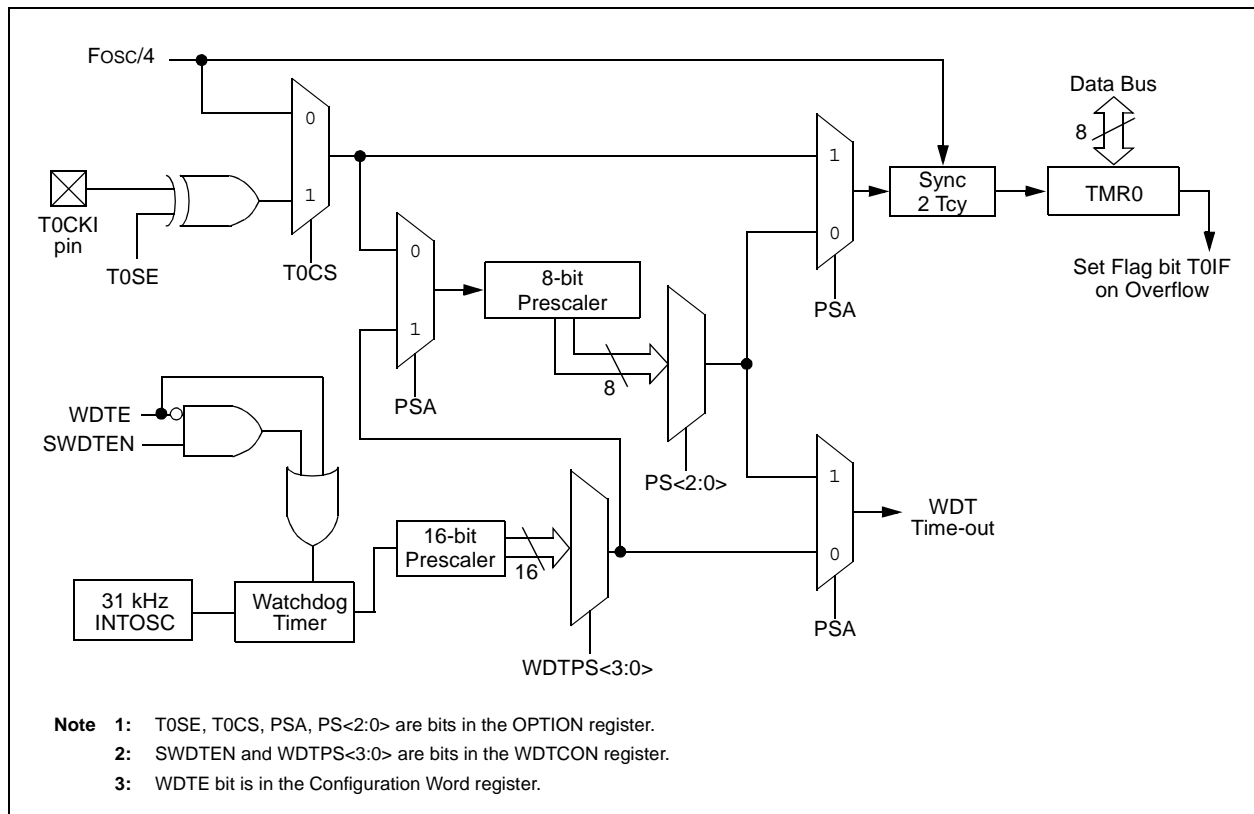
When TMR0 is written, the increment is inhibited for two instruction cycles immediately following the write.

**Note:** The value written to the TMR0 register can be adjusted, in order to account for the two instruction cycle delay when TMR0 is written.

### 5.1.2 8-BIT COUNTER MODE

When used as a counter, the Timer0 module will increment on every rising or falling edge of the T0CKI pin. The incrementing edge is determined by the T0SE bit of the OPTION register. Counter mode is selected by setting the T0CS bit of the OPTION register to '1'.

**FIGURE 5-1: BLOCK DIAGRAM OF THE TIMER0/WDT PRESCALER**



## 5.1.3 SOFTWARE PROGRAMMABLE PRESCALER

A single software programmable prescaler is available for use with either Timer0 or the Watchdog Timer (WDT), but not both simultaneously. The prescaler assignment is controlled by the PSA bit of the OPTION register. To assign the prescaler to Timer0, the PSA bit must be cleared to a '0'.

There are 8 prescaler options for the Timer0 module ranging from 1:2 to 1:256. The prescale values are selectable via the PS<2:0> bits of the OPTION register. In order to have a 1:1 prescaler value for the Timer0 module, the prescaler must be assigned to the WDT module.

The prescaler is not readable or writable. When assigned to the Timer0 module, all instructions writing to the TMR0 register will clear the prescaler.

When the prescaler is assigned to WDT, a CLRWDI instruction will clear the prescaler along with the WDT.

### 5.1.3.1 Switching Prescaler Between Timer0 and WDT Modules

As a result of having the prescaler assigned to either Timer0 or the WDT, it is possible to generate an unintended device Reset when switching prescaler values. When changing the prescaler assignment from Timer0 to the WDT module, the instruction sequence shown in Example 5-1, must be executed.

#### EXAMPLE 5-1: CHANGING PRESCALER (TIMER0 → WDT)

```
BANKSEL    TMR0          ;
CLRWDI     ;Clear WDT
CLRWF     TMR0          ;Clear TMR0 and
                    ;prescaler

BANKSEL    OPTION_REG    ;
BSF        OPTION_REG,PSA ;Select WDT
CLRWDI     ;
                    ;

MOVLW     b'11111000'    ;Mask prescaler
ANDWF     OPTION_REG,W   ;bits
IORLW     b'00000101'    ;Set WDT prescaler
MOVWF     OPTION_REG     ;to 1:32
```

When changing the prescaler assignment from the WDT to the Timer0 module, the following instruction sequence must be executed (see Example 5-2).

#### EXAMPLE 5-2: CHANGING PRESCALER (WDT → TIMER0)

```
CLRWDI     ;Clear WDT and
                    ;prescaler

BANKSEL    OPTION_REG    ;
MOVLW     b'11110000'    ;Mask TMR0 select and
ANDWF     OPTION_REG,W   ;prescaler bits
IORLW     b'00000011'    ;Set prescale to 1:16
MOVWF     OPTION_REG     ;
```

## 5.1.4 TIMER0 INTERRUPT

Timer0 will generate an interrupt when the TMR0 register overflows from FFh to 00h. The T0IF interrupt flag bit of the INTCON register is set every time the TMR0 register overflows, regardless of whether or not the Timer0 interrupt is enabled. The T0IF bit must be cleared in software. The Timer0 interrupt enable is the T0IE bit of the INTCON register.

**Note:** The Timer0 interrupt cannot wake the processor from Sleep since the timer is frozen during Sleep.

## 5.1.5 USING TIMER0 WITH AN EXTERNAL CLOCK

When Timer0 is in Counter mode, the synchronization of the T0CKI input and the Timer0 register is accomplished by sampling the prescaler output on the Q2 and Q4 cycles of the internal phase clocks. Therefore, the high and low periods of the external clock source must meet the timing requirements as shown in the Section 15.0 "Electrical Specifications".

## REGISTER 5-1: OPTION\_REG: OPTION REGISTER

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
$\overline{\text{GPPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
bit 7							bit 0

### Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

- bit 7  **$\overline{\text{GPPU}}$** : GPIO Pull-up Enable bit  
 1 = GPIO pull-ups are disabled  
 0 = GPIO pull-ups are enabled by individual PORT latch values in WPU register
- bit 6 **INTEDG**: Interrupt Edge Select bit  
 1 = Interrupt on rising edge of INT pin  
 0 = Interrupt on falling edge of INT pin
- bit 5 **T0CS**: Timer0 Clock Source Select bit  
 1 = Transition on T0CKI pin  
 0 = Internal instruction cycle clock (Fosc/4)
- bit 4 **T0SE**: Timer0 Source Edge Select bit  
 1 = Increment on high-to-low transition on T0CKI pin  
 0 = Increment on low-to-high transition on T0CKI pin
- bit 3 **PSA**: Prescaler Assignment bit  
 1 = Prescaler is assigned to the WDT  
 0 = Prescaler is assigned to the Timer0 module
- bit 2-0 **PS<2:0>**: Prescaler Rate Select bits

BIT VALUE	TIMER0 RATE	WDT RATE
000	1 : 2	1 : 1
001	1 : 4	1 : 2
010	1 : 8	1 : 4
011	1 : 16	1 : 8
100	1 : 32	1 : 16
101	1 : 64	1 : 32
110	1 : 128	1 : 64
111	1 : 256	1 : 128

**Note 1:** A dedicated 16-bit WDT postscaler is available. See **Section 12.6 “Watchdog Timer (WDT)”** for more information.

**TABLE 5-1: SUMMARY OF REGISTERS ASSOCIATED WITH TIMER0**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
TMR0	Timer0 Module Register								xxxx xxxx	uuuu uuuu
INTCON	GIE	PEIE	T0IE	INTE	GPIE	T0IF	INTF	GPIF	0000 0000	0000 000x
OPTION_REG	$\overline{\text{GPPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
TRISIO	—	—	TRISIO5	TRISIO4	TRISIO3	TRISIO2	TRISIO1	TRISIO0	--11 1111	--11 1111

**Legend:** — = Unimplemented locations, read as '0', u = unchanged, x = unknown. Shaded cells are not used by the Timer0 module.

# PIC12F683

## 6.0 TIMER1 MODULE WITH GATE CONTROL

The Timer1 module is a 16-bit timer/counter with the following features:

- 16-bit timer/counter register pair (TMR1H:TMR1L)
- Programmable internal or external clock source
- 3-bit prescaler
- Optional LP oscillator
- Synchronous or asynchronous operation
- Timer1 gate (count enable) via comparator or  $\overline{T1G}$  pin
- Interrupt on overflow
- Wake-up on overflow (external clock, Asynchronous mode only)
- Special Event Trigger (with CCP)
- Comparator output synchronization to Timer1 clock

Figure 6-1 is a block diagram of the Timer1 module.

## 6.1 Timer1 Operation

The Timer1 module is a 16-bit incrementing counter which is accessed through the TMR1H:TMR1L register pair. Writes to TMR1H or TMR1L directly update the counter.

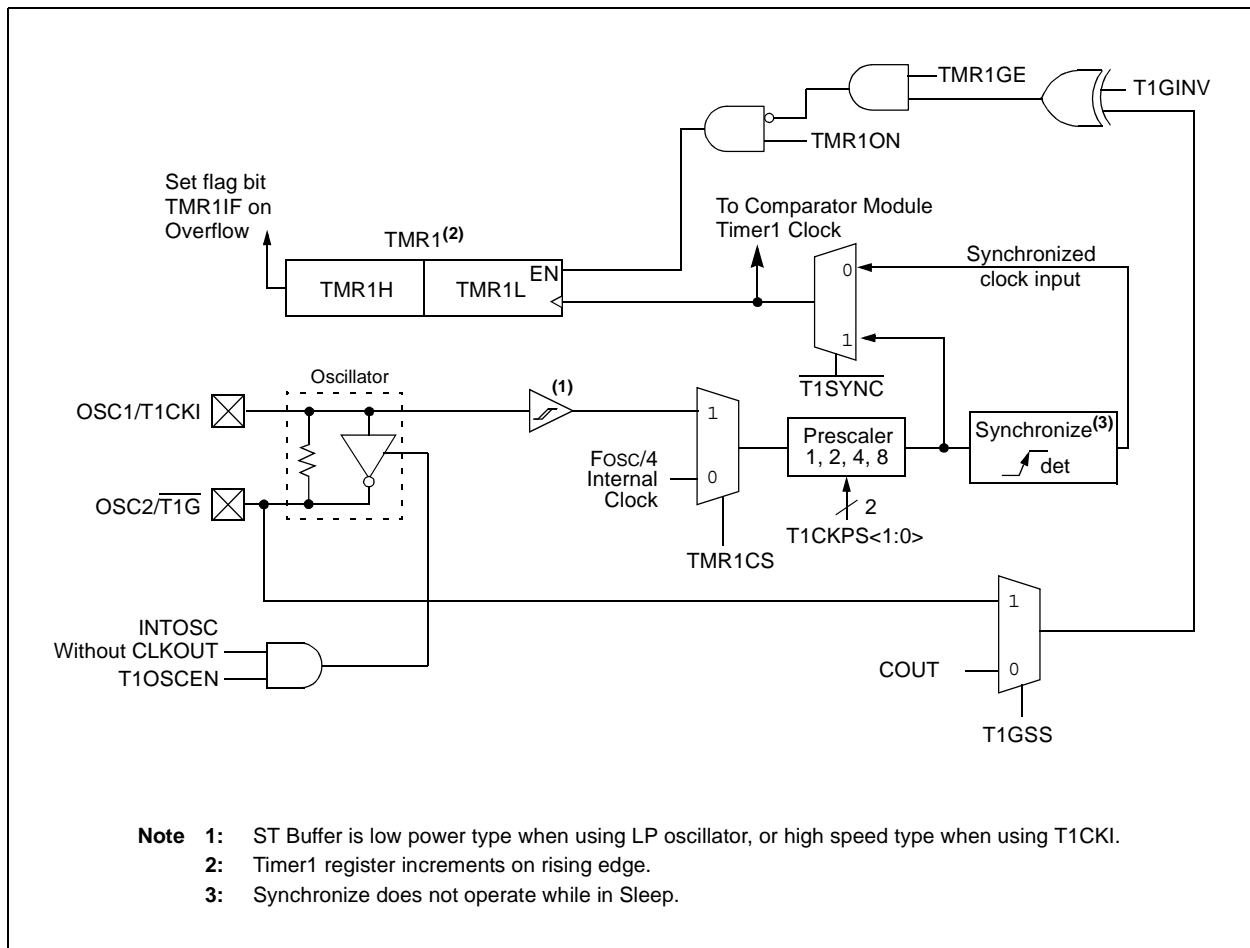
When used with an internal clock source, the module is a timer. When used with an external clock source, the module can be used as either a timer or counter.

## 6.2 Clock Source Selection

The TMR1CS bit of the T1CON register is used to select the clock source. When TMR1CS = 0, the clock source is  $F_{osc}/4$ . When TMR1CS = 1, the clock source is supplied externally.

Clock Source	TMR1CS
$F_{osc}/4$	0
T1CKI pin	1

**FIGURE 6-1: TIMER1 BLOCK DIAGRAM**





## 6.2.1 INTERNAL CLOCK SOURCE

When the internal clock source is selected the TMR1H:TMR1L register pair will increment on multiples of Tcy as determined by the Timer1 prescaler.

## 6.2.2 EXTERNAL CLOCK SOURCE

When the external clock source is selected, the Timer1 module may work as a timer or a counter.

When counting, Timer1 is incremented on the rising edge of the external clock input T1CKI. In addition, the Counter mode clock can be synchronized to the microcontroller system clock or run asynchronously.

If an external clock oscillator is needed (and the microcontroller is using the INTOSC without CLKOUT), Timer1 can use the LP oscillator as a clock source.

**Note:** In Counter mode, a falling edge must be registered by the counter prior to the first incrementing rising edge.

## 6.3 Timer1 Prescaler

Timer1 has four prescaler options allowing 1, 2, 4 or 8 divisions of the clock input. The T1CKPS bits of the T1CON register control the prescale counter. The prescale counter is not directly readable or writable; however, the prescaler counter is cleared upon a write to TMR1H or TMR1L.

## 6.4 Timer1 Oscillator

A low-power 32.768 kHz crystal oscillator is built-in between pins OSC1 (input) and OSC2 (amplifier output). The oscillator is enabled by setting the T1OSCEN control bit of the T1CON register. The oscillator will continue to run during Sleep.

The Timer1 oscillator is shared with the system LP oscillator. Thus, Timer1 can use this mode only when the primary system clock is derived from the internal oscillator or when in LP oscillator mode. The user must provide a software time delay to ensure proper oscillator start-up.

TRISIO<5:4> bits are set when the Timer1 oscillator is enabled. GP5 and GP4 bits read as '0' and TRISIO5 and TRISIO4 bits read as '1'.

**Note:** The oscillator requires a start-up and stabilization time before use. Thus, T1OSCEN should be set and a suitable delay observed prior to enabling Timer1.

## 6.5 Timer1 Operation in Asynchronous Counter Mode

If control bit T1SYNC of the T1CON register is set, the external clock input is not synchronized. The timer continues to increment asynchronous to the internal phase clocks. The timer will continue to run during Sleep and can generate an interrupt on overflow, which will wake-up the processor. However, special precautions in software are needed to read/write the timer (see **Section 6.5.1 "Reading and Writing Timer1 in Asynchronous Counter Mode"**).

**Note:** When switching from synchronous to asynchronous operation, it is possible to skip an increment. When switching from asynchronous to synchronous operation, it is possible to produce a single spurious increment.

### 6.5.1 READING AND WRITING TIMER1 IN ASYNCHRONOUS COUNTER MODE

Reading TMR1H or TMR1L while the timer is running from an external asynchronous clock will ensure a valid read (taken care of in hardware). However, the user should keep in mind that reading the 16-bit timer in two 8-bit values itself, poses certain problems, since the timer may overflow between the reads.

For writes, it is recommended that the user simply stop the timer and write the desired values. A write contention may occur by writing to the timer registers, while the register is incrementing. This may produce an unpredictable value in the TMR1H:TMR1L register pair.

## 6.6 Timer1 Gate

Timer1 gate source is software configurable to be the T1G pin or the output of the Comparator. This allows the device to directly time external events using T1G or analog events using Comparator 2. See the CMCON1 register (**Register 8-2**) for selecting the Timer1 gate source. This feature can simplify the software for a Delta-Sigma A/D converter and many other applications. For more information on Delta-Sigma A/D converters, see the Microchip web site ([www.microchip.com](http://www.microchip.com)).

**Note:** TMR1GE bit of the T1CON register must be set to use either T1G or COUT as the Timer1 gate source. See **Register 8-2** for more information on selecting the Timer1 gate source.

Timer1 gate can be inverted using the T1GINV bit of the T1CON register, whether it originates from the T1G pin or Comparator 2 output. This configures Timer1 to measure either the active-high or active-low time between events.

## 6.7 Timer1 Interrupt

The Timer1 register pair (TMR1H:TMR1L) increments to FFFFh and rolls over to 0000h. When Timer1 rolls over, the Timer1 interrupt flag bit of the PIR1 register is set. To enable the interrupt on rollover, you must set these bits:

- Timer1 interrupt enable bit of the PIE1 register
- PEIE bit of the INTCON register
- GIE bit of the INTCON register

The interrupt is cleared by clearing the TMR1IF bit in the Interrupt Service Routine.

**Note:** The TMR1H:TMR1L register pair and the TMR1IF bit should be cleared before enabling interrupts.

## 6.8 Timer1 Operation During Sleep

Timer1 can only operate during Sleep when setup in Asynchronous Counter mode. In this mode, an external crystal or clock source can be used to increment the counter. To set up the timer to wake the device:

- TMR1ON bit of the T1CON register must be set
- TMR1IE bit of the PIE1 register must be set
- PEIE bit of the INTCON register must be set

The device will wake-up on an overflow and execute the next instruction. If the GIE bit of the INTCON register is set, the device will call the Interrupt Service Routine (0004h).

## 6.9 CCP Special Event Trigger

If a CCP is configured to trigger a special event, the trigger will clear the TMR1H:TMR1L register pair. This special event does not cause a Timer1 interrupt. The CCP module may still be configured to generate a CCP interrupt.

In this mode of operation, the CCPR1H:CCPR1L register pair effectively becomes the period register for Timer1.

Timer1 should be synchronized to the FOSC to utilize the Special Event Trigger. Asynchronous operation of Timer1 can cause a Special Event Trigger to be missed.

In the event that a write to TMR1H or TMR1L coincides with a Special Event Trigger from the CCP, the write will take precedence.

For more information, see **Section on CCP**.

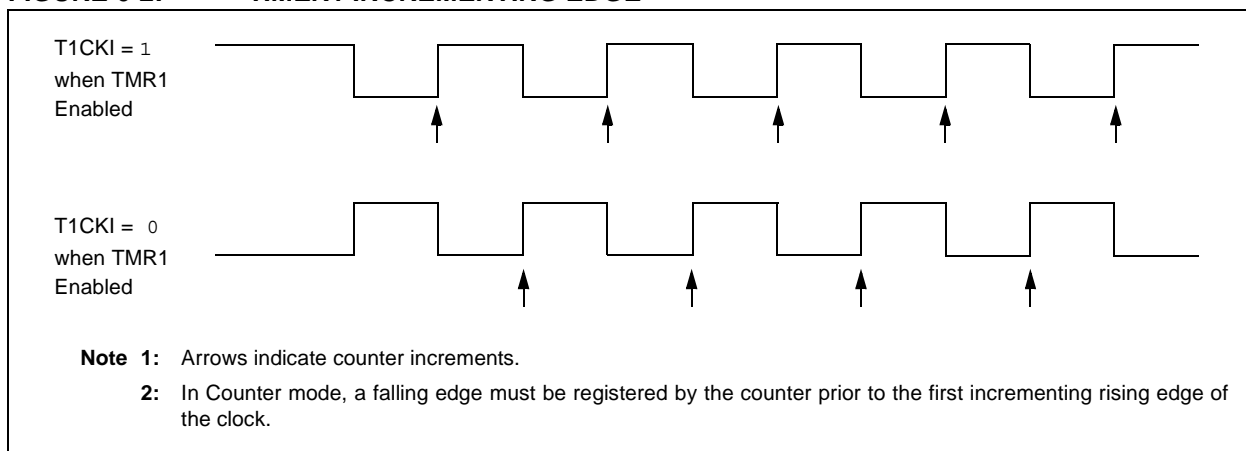
## 6.10 Comparator Synchronization

The same clock used to increment Timer1 can also be used to synchronize the comparator output. This feature is enabled in the Comparator module.

When using the comparator for Timer1 gate, the comparator output should be synchronized to Timer1. This ensures Timer1 does not miss an increment if the comparator changes.

For more information, see **Section 8.0 “Comparator Module”**.

**FIGURE 6-2: TIMER1 INCREMENTING EDGE**



## 6.11 Timer1 Control Register

The Timer1 Control register (T1CON), shown in Register 6-1, is used to control Timer1 and select the various features of the Timer1 module.

**REGISTER 6-1: T1CON: TIMER1 CONTROL REGISTER**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
T1GINV <sup>(1)</sup>	TMR1GE <sup>(2)</sup>	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON
bit 7							bit 0

**Legend:**

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

- bit 7      **T1GINV:** Timer1 Gate Invert bit<sup>(1)</sup>  
             1 = Timer1 gate is active-high (Timer1 counts when gate is high)  
             0 = Timer1 gate is active-low (Timer1 counts when gate is low)
- bit 6      **TMR1GE:** Timer1 Gate Enable bit<sup>(2)</sup>  
             If TMR1ON = 0:  
                 This bit is ignored  
             If TMR1ON = 1:  
                 1 = Timer1 is on if Timer1 gate is not active  
                 0 = Timer1 is on
- bit 5-4    **T1CKPS<1:0>:** Timer1 Input Clock Prescale Select bits  
             11 = 1:8 Prescale Value  
             10 = 1:4 Prescale Value  
             01 = 1:2 Prescale Value  
             00 = 1:1 Prescale Value
- bit 3      **T1OSCEN:** LP Oscillator Enable Control bit  
             If INTOSC without CLKOUT oscillator is active:  
                 1 = LP oscillator is enabled for Timer1 clock  
                 0 = LP oscillator is off  
             Else:  
                 This bit is ignored. LP oscillator is disabled.
- bit 2      **T1SYNC:** Timer1 External Clock Input Synchronization Control bit  
             TMR1CS = 1:  
                 1 = Do not synchronize external clock input  
                 0 = Synchronize external clock input  
             TMR1CS = 0:  
                 This bit is ignored. Timer1 uses the internal clock
- bit 1      **TMR1CS:** Timer1 Clock Source Select bit  
             1 = External clock from T1CKI pin (on the rising edge)  
             0 = Internal clock (FOSC/4)
- bit 0      **TMR1ON:** Timer1 On bit  
             1 = Enables Timer1  
             0 = Stops Timer1

**Note 1:** T1GINV bit inverts the Timer1 gate logic, regardless of source.

**2:** TMR1GE bit must be set to use either  $\overline{T1G}$  pin or COUT, as selected by the T1GSS bit of the CMCON1 register, as a Timer1 gate source.

# PIC12F683

**TABLE 6-1: SUMMARY OF REGISTERS ASSOCIATED WITH TIMER1**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
CONFIG <sup>(1)</sup>	CPD	CP	MCLRE	PWRTÉ	WDTE	FOSC2	FOSC1	FOSC0	—	—
CMCON1	—	—	—	—	—	—	T1GSS	CMSYNC	---- --10	---- --10
INTCON	GIE	PEIE	T0IE	INTE	GPIE	T0IF	INTF	GPIF	0000 0000	0000 000x
PIE1	EEIE	ADIE	CCP1IE	—	CMIE	OSFIE	TMR2IE	TMR1IE	000- 0000	000- 0000
PIR1	EEIF	ADIF	CCP1IF	—	CMIF	OSFIF	TMR2IF	TMR1IF	000- 0000	000- 0000
TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	uuuu uuuu
TMR1L	Holding Register for the Least Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	uuuu uuuu
T1CON	T1GINV	TMR1GE	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON	0000 0000	uuuu uuuu

**Legend:** x = unknown, u = unchanged, - = unimplemented, read as '0'. Shaded cells are not used by the Timer1 module.

**Note 1:** See Configuration Word register (Register 12-1) for operation of all register bits.

## 7.0 TIMER2 MODULE

The Timer2 module is an 8-bit timer with the following features:

- 8-bit timer register (TMR2)
- 8-bit period register (PR2)
- Interrupt on TMR2 match with PR2
- Software programmable prescaler (1:1, 1:4, 1:16)
- Software programmable postscaler (1:1 to 1:16)

See Figure 7-1 for a block diagram of Timer2.

### 7.1 Timer2 Operation

The clock input to the Timer2 module is the system instruction clock ( $F_{osc}/4$ ). The clock is fed into the Timer2 prescaler, which has prescale options of 1:1, 1:4 or 1:16. The output of the prescaler is then used to increment the TMR2 register.

The values of TMR2 and PR2 are constantly compared to determine when they match. TMR2 will increment from 00h until it matches the value in PR2. When a match occurs, two things happen:

- TMR2 is reset to 00h on the next increment cycle.
- The Timer2 postscaler is incremented

The match output of the Timer2/PR2 comparator is then fed into the Timer2 postscaler. The postscaler has postscale options of 1:1 to 1:16 inclusive. The output of the Timer2 postscaler is used to set the TMR2IF interrupt flag bit in the PIR1 register.

The TMR2 and PR2 registers are both fully readable and writable. On any Reset, the TMR2 register is set to 00h and the PR2 register is set to FFh.

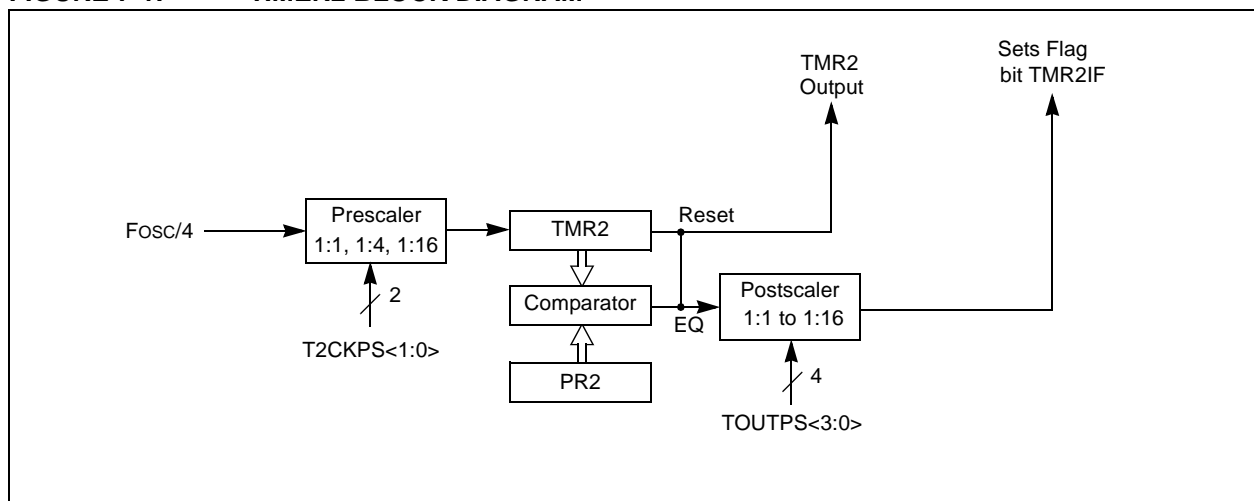
Timer2 is turned on by setting the TMR2ON bit in the T2CON register to a '1'. Timer2 is turned off by clearing the TMR2ON bit to a '0'.

The Timer2 prescaler is controlled by the T2CKPS bits in the T2CON register. The Timer2 postscaler is controlled by the TOUTPS bits in the T2CON register. The prescaler and postscaler counters are cleared when:

- A write to TMR2 occurs.
- A write to T2CON occurs.
- Any device Reset occurs (Power-on Reset,  $\overline{\text{MCLR}}$  Reset, Watchdog Timer Reset, or Brown-out Reset).

**Note:** TMR2 is not cleared when T2CON is written.

**FIGURE 7-1: TIMER2 BLOCK DIAGRAM**



# PIC12F683

**REGISTER 7-1: T2CON: TIMER 2 CONTROL REGISTER**

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0
bit 7							bit 0

**Legend:**

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

bit 7 **Unimplemented:** Read as '0'

bit 6-3 **TOUTPS<3:0>:** Timer2 Output Postscaler Select bits

0000 = 1:1 Postscaler

0001 = 1:2 Postscaler

0010 = 1:3 Postscaler

0011 = 1:4 Postscaler

0100 = 1:5 Postscaler

0101 = 1:6 Postscaler

0110 = 1:7 Postscaler

0111 = 1:8 Postscaler

1000 = 1:9 Postscaler

1001 = 1:10 Postscaler

1010 = 1:11 Postscaler

1011 = 1:12 Postscaler

1100 = 1:13 Postscaler

1101 = 1:14 Postscaler

1110 = 1:15 Postscaler

1111 = 1:16 Postscaler

bit 2 **TMR2ON:** Timer2 On bit

1 = Timer2 is on

0 = Timer2 is off

bit 1-0 **T2CKPS<1:0>:** Timer2 Clock Prescale Select bits

00 = Prescaler is 1

01 = Prescaler is 4

1x = Prescaler is 16

**TABLE 7-1: SUMMARY OF ASSOCIATED TIMER2 REGISTERS**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
INTCON	GIE	PEIE	TOIE	INTE	GPIE	TOIF	INTF	GPIF	0000 0000	0000 000x
PIE1	EEIE	ADIE	CCP1IE	—	CMIE	OSFIE	TMR2IE	TMR1IE	000- 0000	000- 0000
PIR1	EEIF	ADIF	CCP1IF	—	CMIF	OSFIF	TMR2IF	TMR1IF	000- 0000	000- 0000
PR2	Timer2 Module Period Register								1111 1111	1111 1111
TMR2	Holding Register for the 8-bit TMR2 Register								0000 0000	0000 0000
T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	-000 0000

**Legend:** x = unknown, u = unchanged, - = unimplemented read as '0'. Shaded cells are not used for Timer2 module.

## 8.0 COMPARATOR MODULE

Comparators are used to interface analog circuits to a digital circuit by comparing two analog voltages and providing a digital indication of their relative magnitudes. The comparators are very useful mixed signal building blocks because they provide analog functionality independent of the program execution. The analog comparator module includes the following features:

- Multiple comparator configurations
- Comparator output is available internally/externally
- Programmable output polarity
- Interrupt-on-change
- Wake-up from Sleep
- Timer1 gate (count enable)
- Output synchronization to Timer1 clock input
- Programmable voltage reference

### 8.1 Comparator Overview

The comparator is shown in Figure 8-1 along with the relationship between the analog input levels and the digital output. When the analog voltage at VIN+ is less than the analog voltage at VIN-, the output of the comparator is a digital low level. When the analog voltage at VIN+ is greater than the analog voltage at VIN-, the output of the comparator is a digital high level.

FIGURE 8-1: SINGLE COMPARATOR

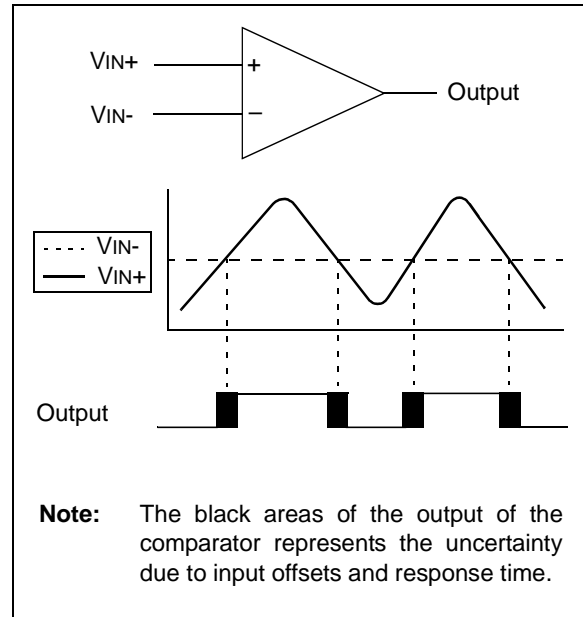
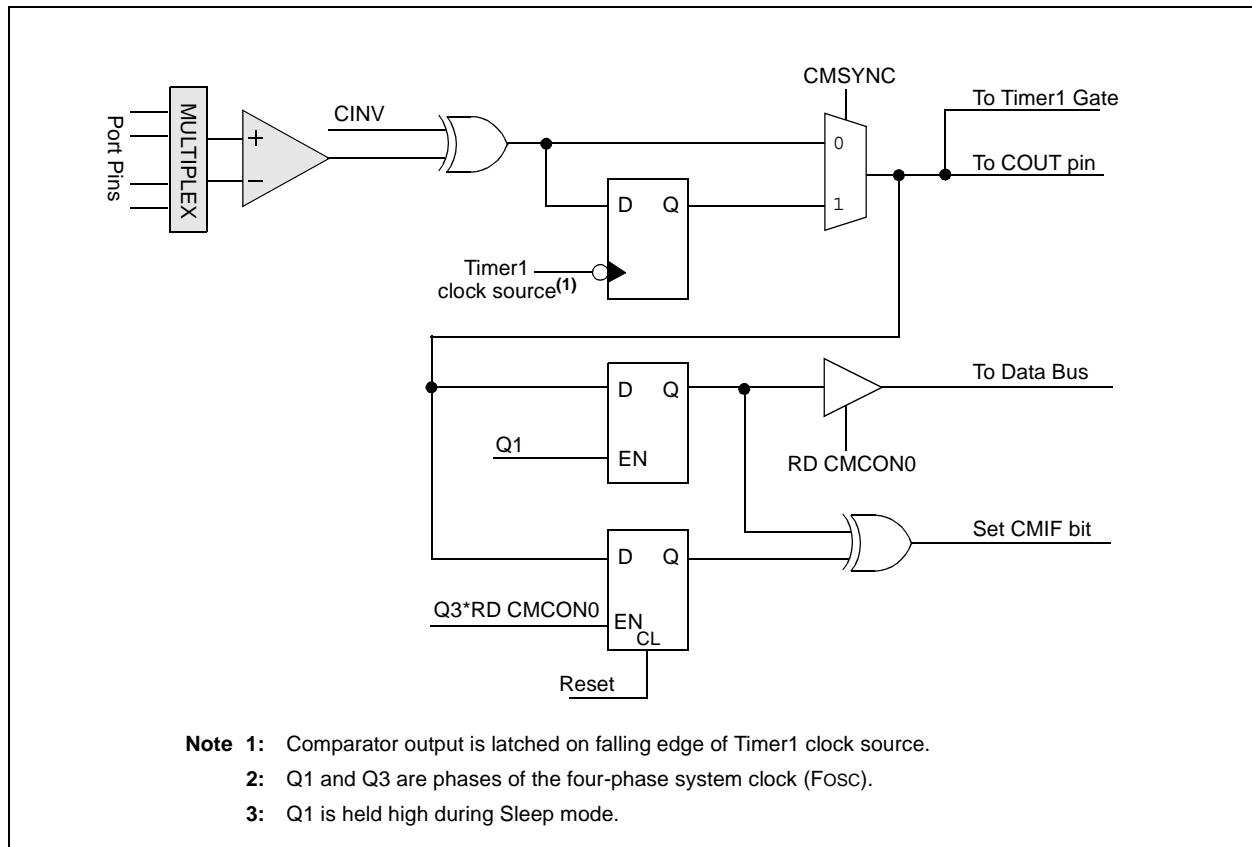


FIGURE 8-2: COMPARATOR OUTPUT BLOCK DIAGRAM



## 8.2 Analog Input Connection Considerations

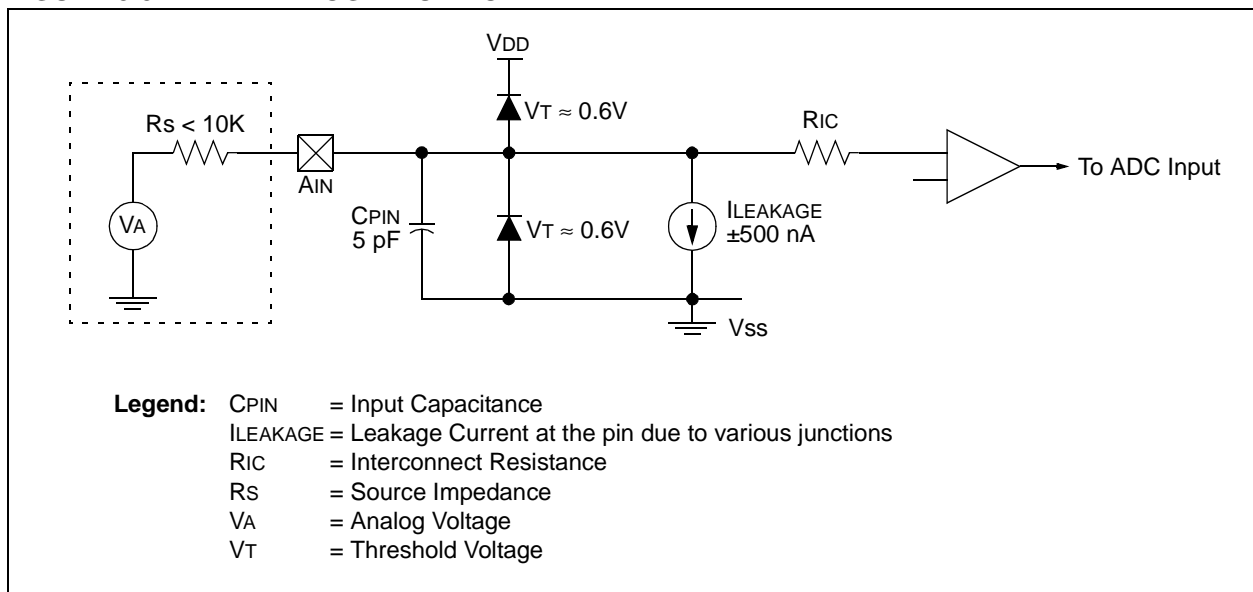
A simplified circuit for an analog input is shown in Figure 8-3. Since the analog input pins share their connection with a digital input, they have reverse biased ESD protection diodes to  $V_{DD}$  and  $V_{SS}$ . The analog input, therefore, must be between  $V_{SS}$  and  $V_{DD}$ . If the input voltage deviates from this range by more than 0.6V in either direction, one of the diodes is forward biased and a latch-up may occur.

A maximum source impedance of 10 k $\Omega$  is recommended for the analog sources. Also, any external component connected to an analog input pin, such as a capacitor or a Zener diode, should have very little leakage current to minimize inaccuracies introduced.

**Note 1:** When reading a PORT register, all pins configured as analog inputs will read as a '0'. Pins configured as digital inputs will convert as an analog input, according to the input specification.

**2:** Analog levels on any pin defined as a digital input, may cause the input buffer to consume more current than is specified.

**FIGURE 8-3: ANALOG INPUT MODEL**





## 8.3 Comparator Configuration

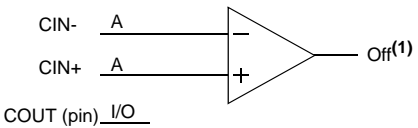
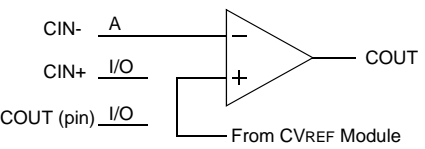
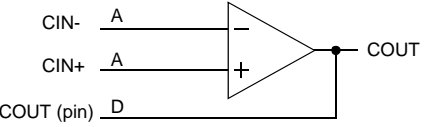
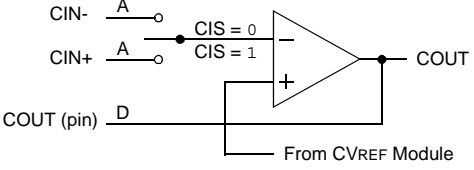
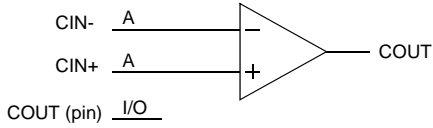
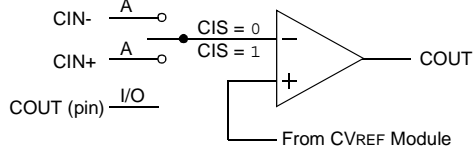
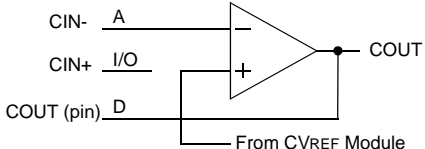
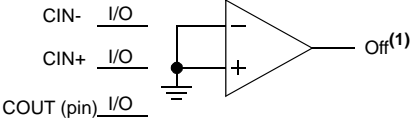
There are eight modes of operation for the comparator. The CM<2:0> bits of the CMCON0 register are used to select these modes as shown in Figure 8-4.

- Analog function (A): digital input buffer is disabled
- Digital function (D): comparator digital output, overrides port function
- Normal port function (I/O): independent of comparator

The port pins denoted as “A” will read as a ‘0’ regardless of the state of the I/O pin or the I/O control TRIS bit. Pins used as analog inputs should also have the corresponding TRIS bit set to ‘1’ to disable the digital output driver. Pins denoted as “D” should have the corresponding TRIS bit set to ‘0’ to enable the digital output driver.

**Note:** Comparator interrupts should be disabled during a Comparator mode change to prevent unintended interrupts.

**FIGURE 8-4: COMPARATOR I/O OPERATING MODES**

<p>Comparator Reset (POR Default Value – low power) CM&lt;2:0&gt; = 000</p> 	<p>Comparator w/o Output and with Internal Reference CM&lt;2:0&gt; = 100</p> 
<p>Comparator with Output CM&lt;2:0&gt; = 001</p> 	<p>Multiplexed Input with Internal Reference and Output CM&lt;2:0&gt; = 101</p> 
<p>Comparator without Output CM&lt;2:0&gt; = 010</p> 	<p>Multiplexed Input with Internal Reference CM&lt;2:0&gt; = 110</p> 
<p>Comparator with Output and Internal Reference CM&lt;2:0&gt; = 011</p> 	<p>Comparator Off (Lowest power) CM&lt;2:0&gt; = 111</p> 
<p><b>Legend:</b> A = Analog Input, ports always reads ‘0’ I/O = Normal port I/O</p> <p><b>Note 1:</b> Reads as ‘0’, unless CINV = 1.</p>	
<p>CIS = Comparator Input Switch (CMCON0&lt;3&gt;) D = Comparator Digital Output</p>	

## 8.4 Comparator Control

The CMCON0 register (Register 8-1) provides access to the following comparator features:

- Mode selection
- Output state
- Output polarity
- Input switch

### 8.4.1 COMPARATOR OUTPUT STATE

The Comparator state can always be read internally via the COUT bit of the CMCON0 register. The comparator state may also be directed to the COUT pin in the following modes:

- CM<2:0> = 001
- CM<2:0> = 011
- CM<2:0> = 101

When one of the above modes is selected, the associated TRIS bit of the COUT pin must be cleared.

### 8.4.2 COMPARATOR OUTPUT POLARITY

Inverting the output of the comparator is functionally equivalent to swapping the comparator inputs. The polarity of the comparator output can be inverted by setting the CINV bit of the CMCON0 register. Clearing CINV results in a non-inverted output. A complete table showing the output state versus input conditions and the polarity bit is shown in Table 8-1.

**TABLE 8-1: OUTPUT STATE VS. INPUT CONDITIONS**

Input Conditions	CINV	COUT
VIN- > VIN+	0	0
VIN- < VIN+	0	1
VIN- > VIN+	1	1
VIN- < VIN+	1	0

**Note:** COUT refers to both the register bit and output pin.

### 8.4.3 COMPARATOR INPUT SWITCH

The inverting input of the comparator may be switched between two analog pins in the following modes:

- CM<2:0> = 101
- CM<2:0> = 110

In the above modes, both pins remain in analog mode regardless of which pin is selected as the input. The CIS bit of the CMCON0 register controls the comparator input switch.

## 8.5 Comparator Response Time

The comparator output is indeterminate for a period of time after the change of an input source or the selection of a new reference voltage. This period is referred to as the response time. The response time of the comparator differs from the settling time of the voltage reference. Therefore, both of these times must be considered when determining the total response time to a comparator input change. See the Comparator and Voltage Reference Specifications in **Section 15.0 “Electrical Specifications”** for more details.

## 8.6 Comparator Interrupt Operation

The comparator interrupt flag is set whenever there is a change in the output value of the comparator. Changes are recognized by means of a mismatch circuit which consists of two latches and an exclusive-or gate (see Figure 8.2). One latch is updated with the comparator output level when the CMCON0 register is read. This latch retains the value until the next read of the CMCON0 register or the occurrence of a Reset. The other latch of the mismatch circuit is updated on every Q1 system clock. A mismatch condition will occur when a comparator output change is clocked through the second latch on the Q1 clock cycle. The mismatch condition will persist, holding the CMIF bit of the PIR1 register true, until either the CMCON0 register is read or the comparator output returns to the previous state.

**Note:** A write operation to the CMCON0 register will also clear the mismatch condition because all writes include a read operation at the beginning of the write cycle.

Software will need to maintain information about the status of the comparator output to determine the actual change that has occurred.

The CMIF bit of the PIR1 register, is the comparator interrupt flag. This bit must be reset in software by clearing it to '0'. Since it is also possible to write a '1' to this register, a simulated interrupt may be initiated.

The CMIE bit of the PIE1 register and the PEIE and GIE bits of the INTCON register must all be set to enable comparator interrupts. If any of these bits are cleared, the interrupt is not enabled, although the CMIF bit of the PIR1 register will still be set if an interrupt condition occurs.

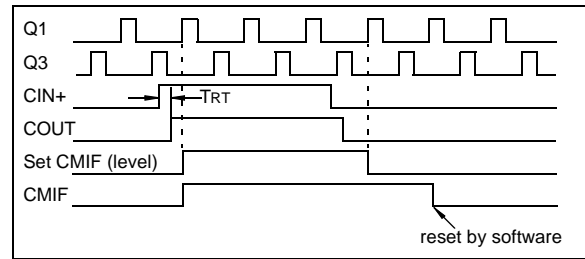
The user, in the Interrupt Service Routine, can clear the interrupt in the following manner:

- a) Any read or write of CMCON0. This will end the mismatch condition.
- b) Clear the CMIF interrupt flag.

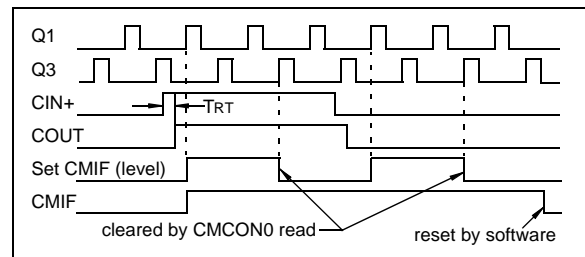
A persistent mismatch condition will preclude clearing the CMIF interrupt flag. Reading CMCON0 will end the mismatch condition and allow the CMIF bit to be cleared.

**Note:** If a change in the CMCON0 register (COUT) should occur when a read operation is being executed (start of the Q2 cycle), then the CMIF interrupt flag may not get set.

**FIGURE 8-5: COMPARATOR INTERRUPT TIMING W/O CMCON0 READ**



**FIGURE 8-6: COMPARATOR INTERRUPT TIMING WITH CMCON0 READ**



**Note 1:** If a change in the CMCON0 register (COUT) should occur when a read operation is being executed (start of the Q2 cycle), then the CMIF of the PIR1 register interrupt flag may not get set.

- 2: When either comparator is first enabled, bias circuitry in the Comparator module may cause an invalid output from the comparator until the bias circuitry is stable. Allow about 1  $\mu$ s for bias settling then clear the mismatch condition and interrupt flags before enabling comparator interrupts.

# PIC12F683

## 8.7 Operation During Sleep

The comparator, if enabled before entering Sleep mode, remains active during Sleep. The additional current consumed by the comparator is shown separately in **Section 15.0 “Electrical Specifications”**. If the comparator is not used to wake the device, power consumption can be minimized while in Sleep mode by turning off the comparator. The comparator is turned off by selecting mode  $CM<2:0> = 000$  or  $CM<2:0> = 111$  of the CMCON0 register.

A change to the comparator output can wake-up the device from Sleep. To enable the comparator to wake the device from Sleep, the CMIE bit of the PIE1 register and the PEIE bit of the INTCON register must be set. The instruction following the Sleep instruction always executes following a wake from Sleep. If the GIE bit of the INTCON register is also set, the device will then execute the Interrupt Service Routine.

## 8.8 Effects of a Reset

A device Reset forces the CMCON0 and CMCON1 registers to their Reset states. This forces the Comparator module to be in the Comparator Reset mode ( $CM<2:0> = 000$ ). Thus, all comparator inputs are analog inputs with the comparator disabled to consume the smallest current possible.

### REGISTER 8-1: CMCON0: COMPARATOR CONFIGURATION REGISTER

U-0	R-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	COUT	—	CINV	CIS	CM2	CM1	CM0
bit 7							bit 0

#### Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

bit 7 **Unimplemented:** Read as '0'

bit 6 **COUT:** Comparator Output bit

When CINV = 0:

1 =  $V_{IN+} > V_{IN-}$

0 =  $V_{IN+} < V_{IN-}$

When CINV = 1:

1 =  $V_{IN+} < V_{IN-}$

0 =  $V_{IN+} > V_{IN-}$

bit 5 **Unimplemented:** Read as '0'

bit 4 **CINV:** Comparator Output Inversion bit

1 = Output inverted

0 = Output not inverted

bit 3 **CIS:** Comparator Input Switch bit

When  $CM<2:0> = 110$  or  $101$ :

1 = CIN+ connects to  $V_{IN-}$

0 = CIN- connects to  $V_{IN-}$

When  $CM<2:0> = 0xx$  or  $100$  or  $111$ :

CIS has no effect.

bit 2-0 **CM<2:0>:** Comparator Mode bits (See Figure 8-5)

000 = CIN pins are configured as analog, COUT pin configured as I/O, Comparator output turned off

001 = CIN pins are configured as analog, COUT pin configured as Comparator output

010 = CIN pins are configured as analog, COUT pin configured as I/O, Comparator output available internally

011 = CIN- pin is configured as analog, CIN+ pin is configured as I/O, COUT pin configured as

Comparator output, CVREF is non-inverting input

100 = CIN- pin is configured as analog, CIN+ pin is configured as I/O, COUT pin is configured as I/O, Comparator output

available internally, CVREF is non-inverting input

101 = CIN pins are configured as analog and multiplexed, COUT pin is configured as

Comparator output, CVREF is non-inverting input

110 = CIN pins are configured as analog and multiplexed, COUT pin is configured as I/O,

Comparator output available internally, CVREF is non-inverting input

111 = CIN pins are configured as I/O, COUT pin is configured as I/O, Comparator output disabled, Comparator off.

## 8.9 Comparator Gating Timer1

This feature can be used to time the duration or interval of analog events. Clearing the T1GSS bit of the CMCON1 register will enable Timer1 to increment based on the output of the comparator. This requires that Timer1 is on and gating is enabled. See **Section 6.0 “Timer1 Module with Gate Control”** for details.

It is recommended to synchronize the comparator with Timer1 by setting the CMSYNC bit when the comparator is used as the Timer1 gate source. This ensures Timer1 does not miss an increment if the comparator changes during an increment.

## 8.10 Synchronizing Comparator Output to Timer1

The comparator output can be synchronized with Timer1 by setting the CMSYNC bit of the CMCON1 register. When enabled, the comparator output is latched on the falling edge of the Timer1 clock source. If a prescaler is used with Timer1, the comparator output is latched after the prescaling function. To prevent a race condition, the comparator output is latched on the falling edge of the Timer1 clock source and Timer1 increments on the rising edge of its clock source. See the Comparator Block Diagram (Figure 8-2) and the Timer1 Block Diagram (Figure 6-1) for more information.

**REGISTER 8-2: CMCON1: COMPARATOR CONFIGURATION REGISTER**

U-0	U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0
—	—	—	—	—	—	T1GSS	CMSYNC
bit 7						bit 0	

**Legend:**

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

bit 7-2 **Unimplemented:** Read as '0'

bit 1 **T1GSS:** Timer1 Gate Source Select bit<sup>(1)</sup>

1 = Timer 1 Gate Source is  $\overline{\text{T1G}}$  pin (pin should be configured as digital input)

0 = Timer 1 Gate Source is comparator output

bit 0 **CMSYNC:** Comparator Output Synchronization bit<sup>(2)</sup>

1 = Output is synchronized with falling edge of Timer1 clock

0 = Output is asynchronous

**Note 1:** Refer to **Section 6.6 “Timer1 Gate”**.

**2:** Refer to Figure 8-2.

# PIC12F683

## 8.11 Comparator Voltage Reference

The Comparator Voltage Reference module provides an internally generated voltage reference for the comparators. The following features are available:

- Independent from Comparator operation
- Two 16-level voltage ranges
- Output clamped to VSS
- Ratiometric with VDD

The VRCON register (Register 8-3) controls the Voltage Reference module shown in Figure 8-7.

### 8.11.1 INDEPENDENT OPERATION

The comparator voltage reference is independent of the comparator configuration. Setting the VREN bit of the VRCON register will enable the voltage reference.

### 8.11.2 OUTPUT VOLTAGE SELECTION

The CVREF voltage reference has 2 ranges with 16 voltage levels in each range. Range selection is controlled by the VRR bit of the VRCON register. The 16 levels are set with the VR<3:0> bits of the VRCON register.

The CVREF output voltage is determined by the following equations:

### EQUATION 8-1: CVREF OUTPUT VOLTAGE

$$\begin{aligned} VRR &= 1 \text{ (low range):} \\ CVREF &= (VR<3:0>/24) \times VDD \\ VRR &= 0 \text{ (high range):} \\ CVREF &= (VDD/4) + (VR<3:0> \times VDD/32) \end{aligned}$$

The full range of VSS to VDD cannot be realized due to the construction of the module. See Figure 8-1.

### 8.11.3 OUTPUT CLAMPED TO VSS

The CVREF output voltage can be set to VSS with no power consumption by configuring VRCON as follows:

- VREN = 0
- VRR = 1
- VR<3:0> = 0000

This allows the comparator to detect a zero-crossing while not consuming additional CVREF module current.

### 8.11.4 OUTPUT RATIOMETRIC TO VDD

The comparator voltage reference is VDD derived and therefore, the CVREF output changes with fluctuations in VDD. The tested absolute accuracy of the Comparator Voltage Reference can be found in **Section 15.0 “Electrical Specifications”**.

## REGISTER 8-3: VRCON: VOLTAGE REFERENCE CONTROL REGISTER

R/W-0	U-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
VREN	—	VRR	—	VR3	VR2	VR1	VR0
bit 7							bit 0

### Legend:

R = Readable bit      W = Writable bit      U = Unimplemented bit, read as ‘0’  
 -n = Value at POR      ‘1’ = Bit is set      ‘0’ = Bit is cleared      x = Bit is unknown

- bit 7      **VREN:** CVREF Enable bit  
           1 = CVREF circuit powered on  
           0 = CVREF circuit powered down, no IDD drain and CVREF = VSS.
- bit 6      **Unimplemented:** Read as ‘0’
- bit 5      **VRR:** CVREF Range Selection bit  
           1 = Low range  
           0 = High range
- bit 4      **Unimplemented:** Read as ‘0’
- bit 3-0    **VR<3:0>:** CVREF Value Selection  $0 \leq VR<3:0> \leq 15$   
           When VRR = 1: CVREF = (VR<3:0>/24) \* VDD  
           When VRR = 0: CVREF = VDD/4 + (VR<3:0>/32) \* VDD

FIGURE 8-7: COMPARATOR VOLTAGE REFERENCE BLOCK DIAGRAM

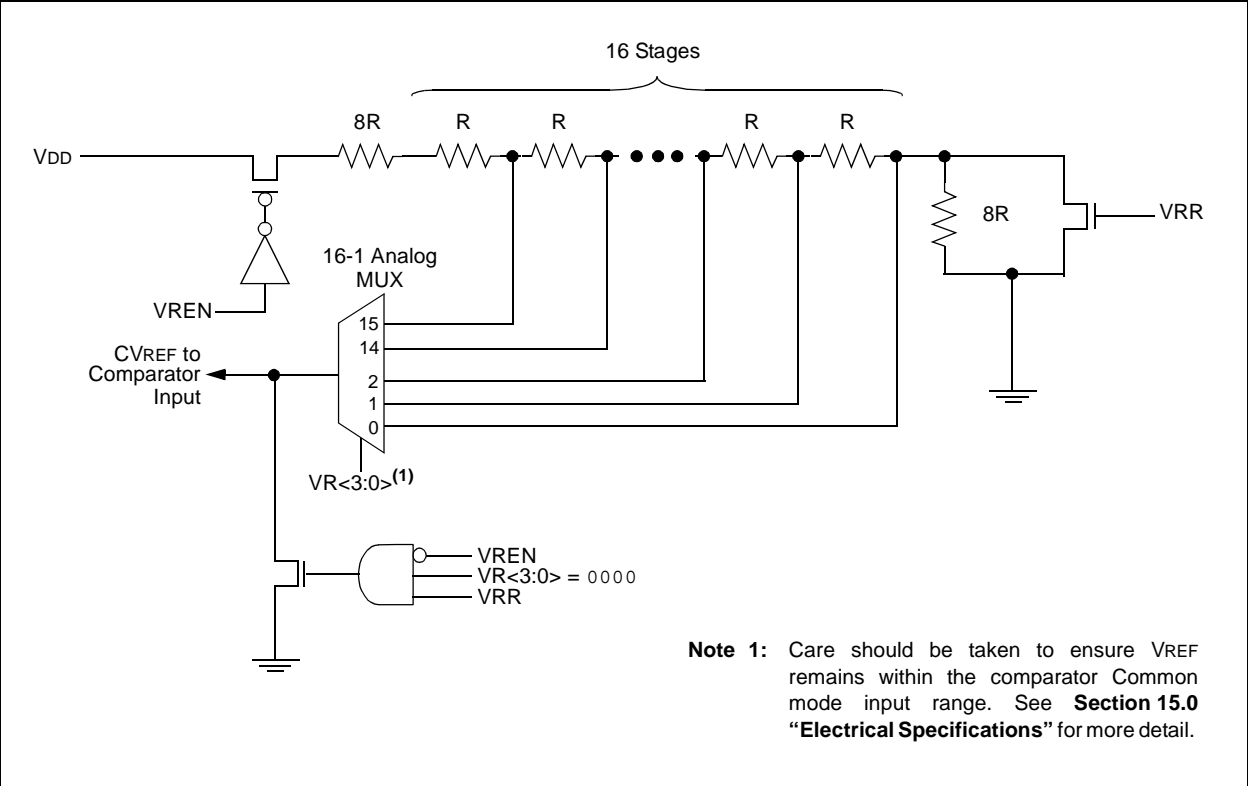


TABLE 8-2: SUMMARY OF REGISTERS ASSOCIATED WITH THE COMPARATOR AND VOLTAGE REFERENCE MODULES

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
ANSEL	—	ADCS2	ADCS1	ADCS0	ANS3	ANS2	ANS1	ANS0	-000 1111	-000 1111
CMCON0	—	COUT	—	CINV	CIS	CM2	CM1	CM0	-0-0 0000	-0-0 0000
CMCON1	—	—	—	—	—	—	T1GSS	CMSYNC	---- --10	---- --10
INTCON	GIE	PEIE	T0IE	INTE	GPIE	T0IF	INTF	GPIF	0000 0000	0000 000x
PIE1	EEIE	ADIE	CCP1IE	—	CMIE	OSFIE	TMR2IE	TMR1IE	000- 0000	0000 0000
PIR1	EEIF	ADIF	CCP1IF	—	CMIF	OSFIF	TMR2IF	TMR1IF	000- 0000	000- 0000
GPIO	—	—	GP5	GP4	GP3	GP2	GP1	GP0	--xx xxxx	--uu uuuu
TRISIO	—	—	TRISIO5	TRISIO4	TRISIO3	TRISIO2	TRISIO1	TRISIO0	--11 1111	--11 1111
VRCON	VREN	—	VRR	—	VR3	VR2	VR1	VR0	0-0- 0000	-0-0 0000

Legend: x = unknown, u = unchanged, - = unimplemented, read as '0'. Shaded cells are not used for comparator.

# PIC12F683

---

NOTES:



## 9.0 ANALOG-TO-DIGITAL CONVERTER (ADC) MODULE

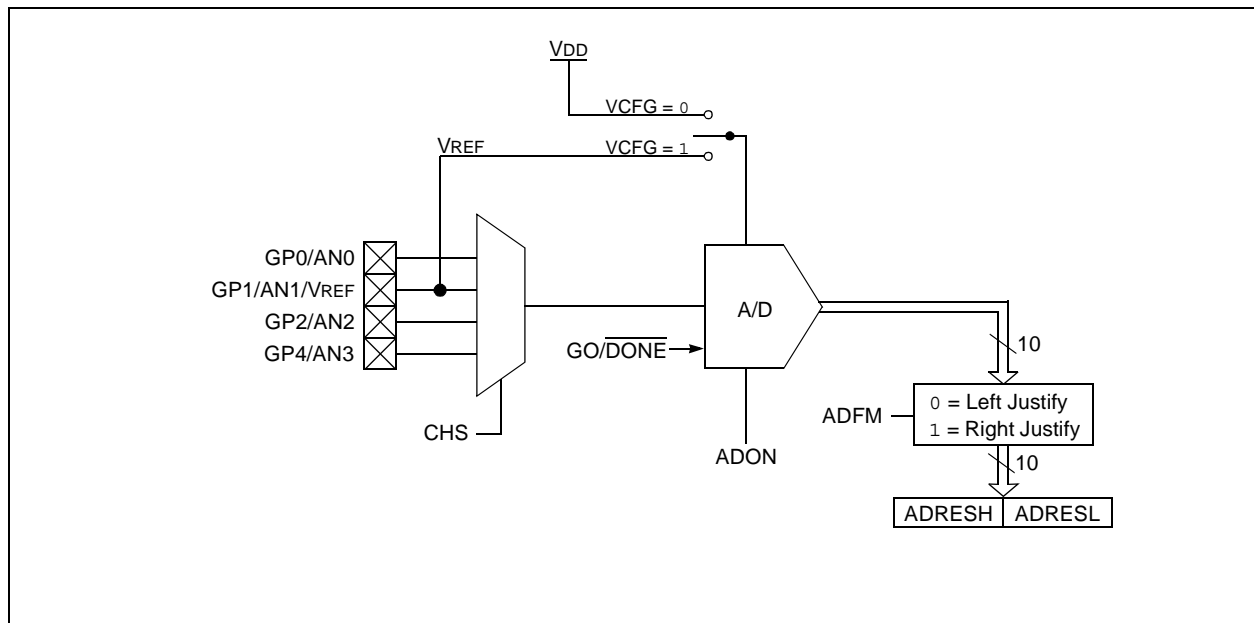
The Analog-to-Digital Converter (ADC) allows conversion of an analog input signal to a 10-bit binary representation of that signal. This device uses analog inputs, which are multiplexed into a single sample and hold circuit. The output of the sample and hold is connected to the input of the converter. The converter generates a 10-bit binary result via successive approximation and stores the conversion result into the ADC result registers (ADRESL and ADRESH).

The ADC voltage reference is software selectable to either VDD or a voltage applied to the external reference pins.

The ADC can generate an interrupt upon completion of a conversion. This interrupt can be used to wake-up the device from Sleep.

Figure 9-1 shows the block diagram of the ADC.

**FIGURE 9-1: ADC BLOCK DIAGRAM**



### 9.1 ADC Configuration

When configuring and using the ADC the following functions must be considered:

- GPIO configuration
- Channel selection
- ADC voltage reference selection
- ADC conversion clock source
- Interrupt control
- Results formatting

#### 9.1.2 CHANNEL SELECTION

The CHS bits of the ADCON0 register determine which channel is connected to the sample and hold circuit.

When changing channels, a delay is required before starting the next conversion. Refer to **Section 9.2 "ADC Operation"** for more information.

#### 9.1.1 GPIO CONFIGURATION

The ADC can be used to convert both analog and digital signals. When converting analog signals, the I/O pin should be configured for analog by setting the associated TRIS and ANSEL bits. See the corresponding GPIO section for more information.

**Note:** Analog voltages on any pin that is defined as a digital input may cause the input buffer to conduct excess current.

# PIC12F683

## 9.1.3 ADC VOLTAGE REFERENCE

The VCFG bit of the ADCON0 register provides control of the positive voltage reference. The positive voltage reference can be either VDD or an external voltage source. The negative voltage reference is always connected to the ground reference.

## 9.1.4 CONVERSION CLOCK

The source of the conversion clock is software selectable via the ADCS bits of the ANSEL register. There are seven possible clock options:

- Fosc/2
- Fosc/4
- Fosc/8
- Fosc/16
- Fosc/32
- Fosc/64
- FRC (dedicated internal oscillator)

The time to complete one bit conversion is defined as TAD. One full 10-bit conversion requires 11 TAD periods as shown in Figure 9-2.

For correct conversion, the appropriate TAD specification must be met. See A/D conversion requirements in **Section 15.0 “Electrical Specifications”** for more information. Table 9-1 gives examples of appropriate ADC clock selections.

**Note:** Unless using the FRC, any changes in the system clock frequency will change the ADC clock frequency, which may adversely affect the ADC result.

**TABLE 9-1: ADC CLOCK PERIOD (TAD) Vs. DEVICE OPERATING FREQUENCIES (VDD ≥ 3.0V)**

ADC Clock Period (TAD)		Device Frequency (Fosc)			
ADC Clock Source	ADCS<2:0>	20 MHz	8 MHz	4 MHz	1 MHz
Fosc/2	000	100 ns <sup>(2)</sup>	250 ns <sup>(2)</sup>	500 ns <sup>(2)</sup>	2.0 μs
Fosc/4	100	200 ns <sup>(2)</sup>	500 ns <sup>(2)</sup>	1.0 μs <sup>(2)</sup>	4.0 μs
Fosc/8	001	400 ns <sup>(2)</sup>	1.0 μs <sup>(2)</sup>	2.0 μs	8.0 μs <sup>(3)</sup>
Fosc/16	101	800 ns <sup>(2)</sup>	2.0 μs	4.0 μs	16.0 μs <sup>(3)</sup>
Fosc/32	010	1.6 μs	4.0 μs	8.0 μs <sup>(3)</sup>	32.0 μs <sup>(3)</sup>
Fosc/64	110	3.2 μs	8.0 μs <sup>(3)</sup>	16.0 μs <sup>(3)</sup>	64.0 μs <sup>(3)</sup>
FRC	x11	2-6 μs <sup>(1,4)</sup>	2-6 μs <sup>(1,4)</sup>	2-6 μs <sup>(1,4)</sup>	2-6 μs <sup>(1,4)</sup>

**Legend:** Shaded cells are outside of recommended range.

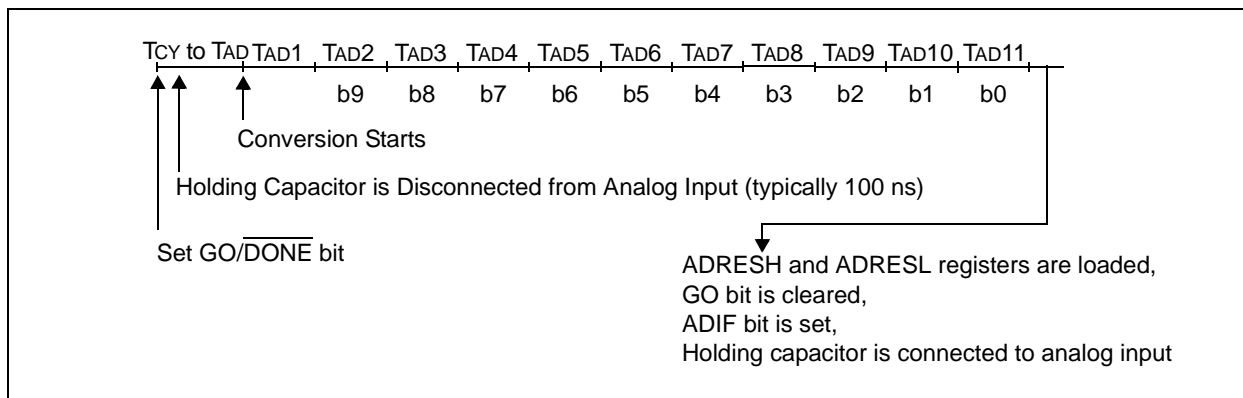
**Note 1:** The FRC source has a typical TAD time of 4 μs for VDD > 3.0V.

**2:** These values violate the minimum required TAD time.

**3:** For faster conversion times, the selection of another clock source is recommended.

**4:** When the device frequency is greater than 1 MHz, the FRC clock source is only recommended if the conversion will be performed during Sleep.

**FIGURE 9-2: ANALOG-TO-DIGITAL CONVERSION TAD CYCLES**



## 9.1.5 INTERRUPTS

The ADC module allows for the ability to generate an interrupt upon completion of an Analog-to-Digital conversion. The ADC interrupt flag is the ADIF bit in the PIR1 register. The ADC interrupt enable is the ADIE bit in the PIE1 register. The ADIF bit must be cleared in software.

**Note:** The ADIF bit is set at the completion of every conversion, regardless of whether or not the ADC interrupt is enabled.

This interrupt can be generated while the device is operating or while in Sleep. If the device is in Sleep, the interrupt will wake-up the device. Upon waking from Sleep, the next instruction following the `SLEEP` instruction is always executed. If the user is attempting to wake-up from Sleep and resume in-line code execution, the global interrupt must be disabled. If the global interrupt is enabled, execution will switch to the interrupt service routine.

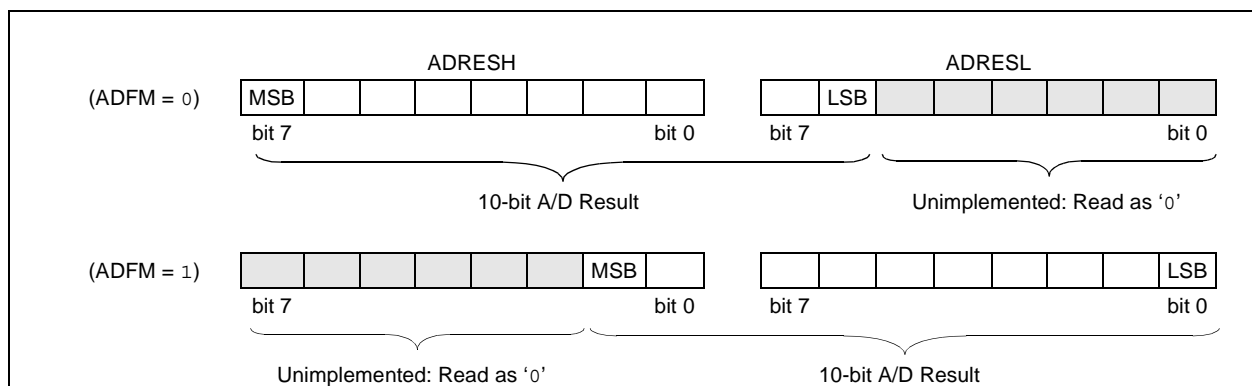
Please see **Section 12.4 “Interrupts”** for more information.

## 9.1.6 RESULT FORMATTING

The 10-bit A/D conversion result can be supplied in two formats, left justified or right justified. The ADFM bit of the ADCON0 register controls the output format.

Figure 9-3 shows the two output formats.

**FIGURE 9-3: 10-BIT A/D CONVERSION RESULT FORMAT**



## 9.2 ADC Operation

### 9.2.1 STARTING A CONVERSION

To enable the ADC module, the ADON bit of the ADCON0 register must be set to a '1'. Setting the `GO/DONE` bit of the ADCON0 register to a '1' will start the Analog-to-Digital conversion.

**Note:** The `GO/DONE` bit should not be set in the same instruction that turns on the ADC. Refer to **Section 9.2.6 “A/D Conversion Procedure”**.

### 9.2.2 COMPLETION OF A CONVERSION

When the conversion is complete, the ADC module will:

- Clear the `GO/DONE` bit
- Set the ADIF flag bit
- Update the ADRESH:ADRESL registers with new conversion result

### 9.2.3 TERMINATING A CONVERSION

If a conversion must be terminated before completion, the `GO/DONE` bit can be cleared in software. The ADRESH:ADRESL registers will not be updated with the partially complete Analog-to-Digital conversion sample. Instead, the ADRESH:ADRESL register pair will retain the value of the previous conversion. Additionally, a 2 TAD delay is required before another acquisition can be initiated. Following this delay, an input acquisition is automatically started on the selected channel.

**Note:** A device Reset forces all registers to their Reset state. Thus, the ADC module is turned off and any pending conversion is terminated.

## 9.2.4 ADC OPERATION DURING SLEEP

The ADC module can operate during Sleep. This requires the ADC clock source to be set to the FRC option. When the FRC clock source is selected, the ADC waits one additional instruction before starting the conversion. This allows the `SLEEP` instruction to be executed, which can reduce system noise during the conversion. If the ADC interrupt is enabled, the device will wake-up from Sleep when the conversion completes. If the ADC interrupt is disabled, the ADC module is turned off after the conversion completes, although the `ADON` bit remains set.

When the ADC clock source is something other than FRC, a `SLEEP` instruction causes the present conversion to be aborted and the ADC module is turned off, although the `ADON` bit remains set.

## 9.2.5 SPECIAL EVENT TRIGGER

The CCP Special Event Trigger allows periodic ADC measurements without software intervention. When this trigger occurs, the `GO/DONE` bit is set by hardware and the Timer1 counter resets to zero.

Using the Special Event Trigger does not assure proper ADC timing. It is the user's responsibility to ensure that the ADC timing requirements are met.

See **Section 11.0 “Capture/Compare/PWM (CCP) Module”** for more information.

## 9.2.6 A/D CONVERSION PROCEDURE

This is an example procedure for using the ADC to perform an Analog-to-Digital conversion:

1. Configure GPIO Port:
  - Disable pin output driver (See `TRIS` register)
  - Configure pin as analog
2. Configure the ADC module:
  - Select ADC conversion clock
  - Configure voltage reference
  - Select ADC input channel
  - Select result format
  - Turn on ADC module
3. Configure ADC interrupt (optional):
  - Clear ADC interrupt flag
  - Enable ADC interrupt
  - Enable peripheral interrupt
  - Enable global interrupt<sup>(1)</sup>
4. Wait the required acquisition time<sup>(2)</sup>.
5. Start conversion by setting the `GO/DONE` bit.
6. Wait for ADC conversion to complete by one of the following:
  - Polling the `GO/DONE` bit
  - Waiting for the ADC interrupt (interrupts enabled)
7. Read ADC Result

8. Clear the ADC interrupt flag (required if interrupt is enabled).

**Note 1:** The global interrupt can be disabled if the user is attempting to wake-up from Sleep and resume in-line code execution.

**2:** See **Section 9.3 “A/D Acquisition Requirements”**.

## EXAMPLE 9-1: A/D CONVERSION

```
;This code block configures the ADC
;for polling, Vdd reference, Frc clock
;and GP0 input.
;
;Conversion start & polling for completion
; are included.
;
BANKSEL  TRISIO      ;
BSF      TRISIO,0    ;Set GP0 to input
BANKSEL  ANSEL       ;
MOVLW    B'01110001' ;ADC Frc clock,
IORWF    ANSEL       ; and GP0 as analog
BANKSEL  ADCON0      ;
MOVLW    B'10000001' ;Right justify,
MOVWF    ADCON0      ;Vdd Vref, AN0, On
CALL     SampleTime  ;Acquisition delay
BSF      ADCON0,GO    ;Start conversion
BTFSF    ADCON0,GO    ;Is conversion done?
GOTO     $-1          ;No, test again
BANKSEL  ADRESH      ;
MOVF     ADRESH,W     ;Read upper 2 bits
MOVWF    RESULTHI     ;Store in GPR space
BANKSEL  ADRESL      ;
MOVF     ADRESL,W     ;Read lower 8 bits
MOVWF    RESULTLO     ;Store in GPR space
```

## 9.2.7 ADC REGISTER DEFINITIONS

The following registers are used to control the operation of the ADC.

## REGISTER 9-1: ADCON0: A/D CONTROL REGISTER 0

R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	VCFG	—	—	CHS1	CHS0	GO/DONE	ADON
bit 7							bit 0

### Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

- bit 7      **ADFM:** A/D Conversion Result Format Select bit  
             1 = Right justified  
             0 = Left justified
- bit 6      **VCFG:** Voltage Reference bit  
             1 = VREF pin  
             0 = VDD
- bit 5-4    **Unimplemented:** Read as '0'
- bit 3-2    **CHS<1:0>:** Analog Channel Select bits  
             00 = AN0  
             01 = AN1  
             10 = AN2  
             11 = AN3
- bit 1      **GO/DONE:** A/D Conversion Status bit  
             1 = A/D conversion cycle in progress. Setting this bit starts an A/D conversion cycle.  
                 This bit is automatically cleared by hardware when the A/D conversion has completed.  
             0 = A/D conversion completed/not in progress
- bit 0      **ADON:** ADC Enable bit  
             1 = ADC is enabled  
             0 = ADC is disabled and consumes no operating current

# PIC12F683

## REGISTER 9-2: ADRESH: ADC RESULT REGISTER HIGH (ADRESH) ADFM = 0

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
ADRES9	ADRES8	ADRES7	ADRES6	ADRES5	ADRES4	ADRES3	ADRES2
bit 7							bit 0

### Legend:

R = Readable bit      W = Writable bit      U = Unimplemented bit, read as '0'  
 -n = Value at POR      '1' = Bit is set      '0' = Bit is cleared      x = Bit is unknown

bit 7-0      **ADRES<9:2>**: ADC Result Register bits  
 Upper 8 bits of 10-bit conversion result

## REGISTER 9-3: ADRESL: ADC RESULT REGISTER LOW (ADRESL) ADFM = 0

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
ADRES1	ADRES0	—	—	—	—	—	—
bit 7							bit 0

### Legend:

R = Readable bit      W = Writable bit      U = Unimplemented bit, read as '0'  
 -n = Value at POR      '1' = Bit is set      '0' = Bit is cleared      x = Bit is unknown

bit 7-6      **ADRES<1:0>**: ADC Result Register bits  
 Lower 2 bits of 10-bit conversion result

bit 5-0      **Reserved**: Do not use.

## REGISTER 9-4: ADRESH: ADC RESULT REGISTER HIGH (ADRESH) ADFM = 1

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	—	—	—	—	ADRES9	ADRES8
bit 7							bit 0

### Legend:

R = Readable bit      W = Writable bit      U = Unimplemented bit, read as '0'  
 -n = Value at POR      '1' = Bit is set      '0' = Bit is cleared      x = Bit is unknown

bit 7-2      **Reserved**: Do not use.

bit 1-0      **ADRES<9:8>**: ADC Result Register bits  
 Upper 2 bits of 10-bit conversion result

## REGISTER 9-5: ADRESL: ADC RESULT REGISTER LOW (ADRESL) ADFM = 1

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
ADRES7	ADRES6	ADRES5	ADRES4	ADRES3	ADRES2	ADRES1	ADRES0
bit 7							bit 0

### Legend:

R = Readable bit      W = Writable bit      U = Unimplemented bit, read as '0'  
 -n = Value at POR      '1' = Bit is set      '0' = Bit is cleared      x = Bit is unknown

bit 7-0      **ADRES<7:0>**: ADC Result Register bits  
 Lower 8 bits of 10-bit conversion result

## 9.3 A/D Acquisition Requirements

For the ADC to meet its specified accuracy, the charge holding capacitor (CHOLD) must be allowed to fully charge to the input channel voltage level. The Analog Input model is shown in Figure 9-4. The source impedance (RS) and the internal sampling switch (RSS) impedance directly affect the time required to charge the capacitor CHOLD. The sampling switch (RSS) impedance varies over the device voltage (VDD), see Figure 9-4. **The maximum recommended impedance for analog sources is 10 kΩ.** As the source impedance is decreased, the acquisition time may be decreased. After the analog input channel is selected (or changed),

an A/D acquisition must be done before the conversion can be started. To calculate the minimum acquisition time, Equation 9-1 may be used. This equation assumes that 1/2 LSb error is used (1024 steps for the ADC). The 1/2 LSb error is the maximum error allowed for the ADC to meet its specified resolution.

### EQUATION 9-1: ACQUISITION TIME EXAMPLE

*Assumptions: Temperature = 50°C and external impedance of 10kΩ 5.0V VDD*

$$\begin{aligned} T_{ACQ} &= \text{Amplifier Settling Time} + \text{Hold Capacitor Charging Time} + \text{Temperature Coefficient} \\ &= T_{AMP} + T_C + T_{COFF} \\ &= 2\mu s + T_C + [(Temperature - 25^\circ C)(0.05\mu s/^\circ C)] \end{aligned}$$

*The value for TC can be approximated with the following equations:*

$$V_{APPLIED} \left( 1 - \frac{1}{2047} \right) = V_{CHOLD} \quad ;[1] \text{ } V_{CHOLD} \text{ charged to within } 1/2 \text{ lsb}$$

$$V_{APPLIED} \left( 1 - e^{\frac{-T_C}{RC}} \right) = V_{CHOLD} \quad ;[2] \text{ } V_{CHOLD} \text{ charge response to } V_{APPLIED}$$

$$V_{APPLIED} \left( 1 - e^{\frac{-T_C}{RC}} \right) = V_{APPLIED} \left( 1 - \frac{1}{2047} \right) \quad ;\text{combining [1] and [2]}$$

*Solving for TC:*

$$\begin{aligned} T_C &= -CHOLD(RIC + RSS + RS) \ln(1/2047) \\ &= -10pF(1k\Omega + 7k\Omega + 10k\Omega) \ln(0.0004885) \\ &= 1.37\mu s \end{aligned}$$

*Therefore:*

$$\begin{aligned} T_{ACQ} &= 2\mu s + 1.37\mu s + [(50^\circ C - 25^\circ C)(0.05\mu s/^\circ C)] \\ &= 4.67\mu s \end{aligned}$$

**Note 1:** The reference voltage (VREF) has no effect on the equation, since it cancels itself out.

**2:** The charge holding capacitor (CHOLD) is not discharged after each conversion.

**3:** The maximum recommended impedance for analog sources is 10 kΩ. This is required to meet the pin leakage specification.

# PIC12F683

FIGURE 9-4: ANALOG INPUT MODEL

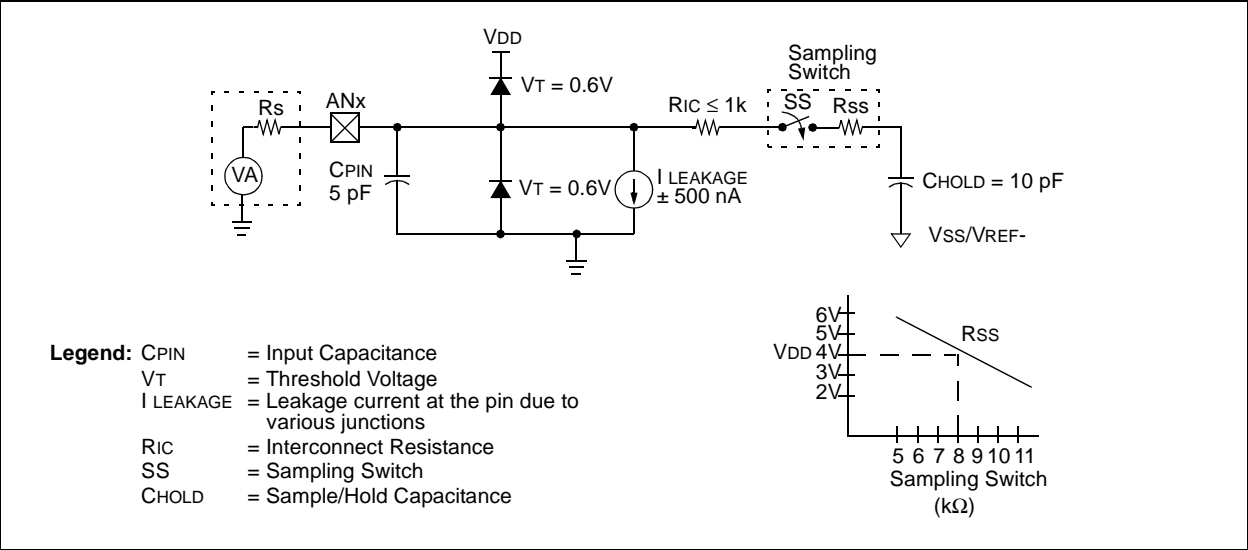
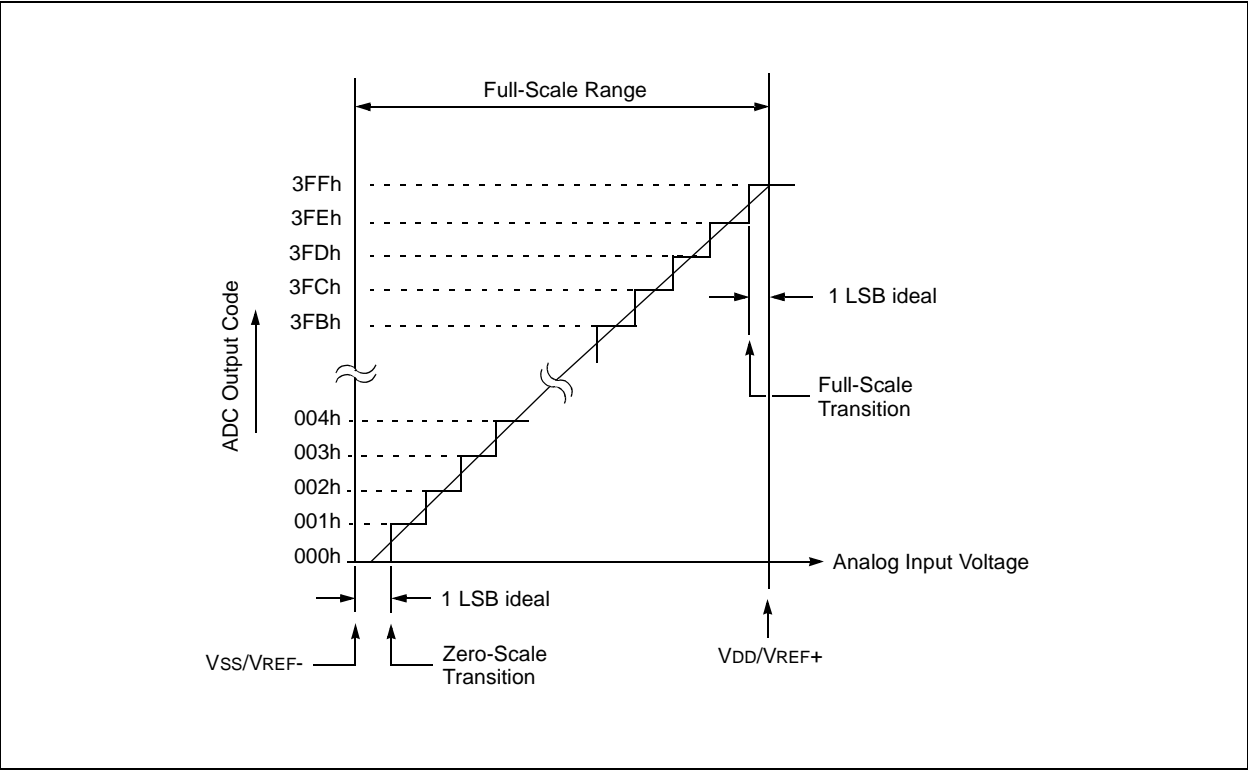


FIGURE 9-5: ADC TRANSFER FUNCTION





**TABLE 9-2: SUMMARY OF ASSOCIATED ADC REGISTERS**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
ADCON0	ADFM	VCFG	—	—	CHS1	CHS0	GO/DONE	ADON	00-- 0000	0000 0000
ANSEL	—	ADCS2	ADCS1	ADCS0	ANS3	ANS2	ANS1	ANS0	-000 1111	-000 1111
ADRESH	A/D Result Register High Byte								xxxx xxxx	uuuu uuuu
ADRESL	A/D Result Register Low Byte								xxxx xxxx	uuuu uuuu
INTCON	GIE	PEIE	T0IE	INTE	GPIE	T0IF	INTF	GPIF	0000 0000	0000 000x
PIE1	EEIE	ADIE	CCP1IE	—	CMIE	OSFIE	TMR2IE	TMR1IE	000- 0000	0000 0000
PIR1	EEIF	ADIF	CCP1IF	—	CMIF	OSFIF	TMR2IF	TMR1IF	000- 0000	000- 0000
GPIO	—	—	GP5	GP4	GP3	GP2	GP1	GP0	--xx xxxx	--uu uuuu
TRISIO	—	—	TRISIO5	TRISIO4	TRISIO3	TRISIO2	TRISIO1	TRISIO0	--11 1111	--11 1111

**Legend:** x = unknown, u = unchanged, — = unimplemented read as '0'. Shaded cells are not used for ADC module.

# PIC12F683

---

NOTES:

## 10.0 DATA EEPROM MEMORY

The EEPROM data memory is readable and writable during normal operation (full VDD range). This memory is not directly mapped in the register file space. Instead, it is indirectly addressed through the Special Function Registers. There are four SFRs used to read and write this memory:

- EECON1
- EECON2 (not a physically implemented register)
- EEDAT
- EEADR

EEDAT holds the 8-bit data for read/write, and EEADR holds the address of the EEPROM location being accessed. PIC12F683 has 256 bytes of data EEPROM with an address range from 0h to FFh.

The EEPROM data memory allows byte read and write. A byte write automatically erases the location and writes the new data (erase before write). The EEPROM data memory is rated for high erase/write cycles. The write time is controlled by an on-chip timer. The write time will vary with voltage and temperature as well as from chip-to-chip. Please refer to AC Specifications in **Section 15.0 “Electrical Specifications”** for exact limits.

When the data memory is code-protected, the CPU may continue to read and write the data EEPROM memory. The device programmer can no longer access the data EEPROM data and will read zeroes.

### REGISTER 10-1: EEDAT: EEPROM DATA REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
EEDAT7	EEDAT6	EEDAT5	EEDAT4	EEDAT3	EEDAT2	EEDAT1	EEDAT0
bit 7							bit 0

#### Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

bit 7-0

**EEDATn**: Byte Value to Write To or Read From Data EEPROM bits

### REGISTER 10-2: EEADR: EEPROM ADDRESS REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
EEADR7	EEADR6	EEADR5	EEADR4	EEADR3	EEADR2	EEADR1	EEADR0
bit 7							bit 0

#### Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

bit 7-0

**EEADR**: Specifies One of 256 Locations for EEPROM Read/Write Operation bits

# PIC12F683

## 10.1 EECON1 and EECON2 Registers

EECON1 is the control register with four low-order bits physically implemented. The upper four bits are non-implemented and read as '0's.

Control bits RD and WR initiate read and write, respectively. These bits cannot be cleared, only set in software. They are cleared in hardware at completion of the read or write operation. The inability to clear the WR bit in software prevents the accidental, premature termination of a write operation.

The WREN bit, when set, will allow a write operation. On power-up, the WREN bit is clear. The WRERR bit is set when a write operation is interrupted by a MCLR Reset, or a WDT Time-out Reset during normal

operation. In these situations, following Reset, the user can check the WRERR bit, clear it and rewrite the location. The data and address will be cleared. Therefore, the EEDAT and EEADR registers will need to be re-initialized.

Interrupt flag, EEIF bit of the PIR1 register, is set when write is complete. This bit must be cleared in software.

EECON2 is not a physical register. Reading EECON2 will read all '0's. The EECON2 register is used exclusively in the data EEPROM write sequence.

**Note:** The EECON1, EEDAT and EEADR registers should not be modified during a data EEPROM write (WR bit = 1).

### REGISTER 10-3: EECON1: EEPROM CONTROL REGISTER

U-0	U-0	U-0	U-0	R/W-x	R/W-0	R/S-0	R/S-0
—	—	—	—	WRERR	WREN	WR	RD
bit 7				bit 0			

#### Legend:

S = Bit can only be set

R = Readable bit

-n = Value at POR

W = Writable bit

'1' = Bit is set

U = Unimplemented bit, read as '0'

'0' = Bit is cleared

x = Bit is unknown

bit 7-4 **Unimplemented:** Read as '0'

bit 3 **WRERR:** EEPROM Error Flag bit

- 1 = A write operation is prematurely terminated (any MCLR Reset, any WDT Reset during normal operation or BOR Reset)
- 0 = The write operation completed

bit 2 **WREN:** EEPROM Write Enable bit

- 1 = Allows write cycles
- 0 = Inhibits write to the data EEPROM

bit 1 **WR:** Write Control bit

- 1 = Initiates a write cycle (The bit is cleared by hardware once write is complete. The WR bit can only be set, not cleared, in software.)
- 0 = Write cycle to the data EEPROM is complete

bit 0 **RD:** Read Control bit

- 1 = Initiates an EEPROM read (Read takes one cycle. RD is cleared in hardware. The RD bit can only be set, not cleared, in software.)
- 0 = Does not initiate an EEPROM read

## 10.2 Reading the EEPROM Data Memory

To read a data memory location, the user must write the address to the EEADR register and then set control bit RD of the EECON1 register, as shown in Example 10-1. The data is available, at the very next cycle, in the EEDAT register. Therefore, it can be read in the next instruction. EEDAT holds this value until another read, or until it is written to by the user (during a write operation).

### EXAMPLE 10-1: DATA EEPROM READ

```
BANKSEL    EEADR    ;
MOVLW      CONFIG_ADDR;
MOVWF      EEADR    ;Address to read
BSF        EECON1,RD ;EE Read
MOVF       EEDAT,W   ;Move data to W
```

## 10.3 Writing to the EEPROM Data Memory

To write an EEPROM data location, the user must first write the address to the EEADR register and the data to the EEDAT register. Then the user must follow a specific sequence to initiate the write for each byte, as shown in Example 10-2.

### EXAMPLE 10-2: DATA EEPROM WRITE

```

Required Sequence
[
BANKSEL    EECON1    ;
BSF        EECON1,WREN ;Enable write
BCF        INTCON,GIE ;Disable INTs
BTFSC      INTCON,GIE ;See AN576
GOTO       $-2        ;
]
MOVLW      55h        ;Unlock write
MOVWF      EECON2      ;
MOVLW      AAh        ;
MOVWF      EECON2      ;
BSF        EECON1,WR   ;Start the write
BSF        INTCON,GIE  ;Enable INTS

```

The write will not initiate if the above sequence is not exactly followed (write 55h to EECON2, write AAh to EECON2, then set WR bit) for each byte. We strongly recommend that interrupts be disabled during this code segment. A cycle count is executed during the required sequence. Any number that is not equal to the required cycles to execute the required sequence will prevent the data from being written into the EEPROM.

Additionally, the WREN bit in EECON1 must be set to enable write. This mechanism prevents accidental writes to data EEPROM due to errant (unexpected) code execution (i.e., lost programs). The user should keep the WREN bit clear at all times, except when updating EEPROM. The WREN bit is not cleared by hardware.

After a write sequence has been initiated, clearing the WREN bit will not affect this write cycle. The WR bit will be inhibited from being set unless the WREN bit is set.

At the completion of the write cycle, the WR bit is cleared in hardware and the EE Write Complete Interrupt Flag bit (EEIF) is set. The user can either enable this interrupt or poll this bit. The EEIF bit of the PIR1 register must be cleared by software.

## 10.4 Write Verify

Depending on the application, good programming practice may dictate that the value written to the data EEPROM should be verified (see Example 10-3) to the desired value to be written.

### EXAMPLE 10-3: WRITE VERIFY

```

BANKSEL    EEDAT      ;
MOVF       EEDAT,W     ;EEDAT not changed
                        ;from previous write
BSF        EECON1,RD   ;YES, Read the
                        ;value written
XORWF      EEDAT,W     ;
BTFSS      STATUS,Z    ;Is data the same
GOTO       WRITE_ERR   ;No, handle error
:           ;Yes, continue

```

### 10.4.1 USING THE DATA EEPROM

The data EEPROM is a high-endurance, byte addressable array that has been optimized for the storage of frequently changing information (e.g., program variables or other data that are updated often). When variables in one section change frequently, while variables in another section do not change, it is possible to exceed the total number of write cycles to the EEPROM (specification D124) without exceeding the total number of write cycles to a single byte (specifications D120 and D120A). If this is the case, then a refresh of the array must be performed. For this reason, variables that change infrequently (such as constants, IDs, calibration, etc.) should be stored in Flash program memory.

# PIC12F683

## 10.5 Protection Against Spurious Write

There are conditions when the user may not want to write to the data EEPROM memory. To protect against spurious EEPROM writes, various mechanisms have been built in. On power-up, WREN is cleared. Also, the Power-up Timer (64 ms duration) prevents EEPROM write.

The write initiate sequence and the WREN bit together help prevent an accidental write during:

- Brown-out
- Power Glitch
- Software Malfunction

## 10.6 Data EEPROM Operation During Code-Protect

Data memory can be code-protected by programming the  $\overline{\text{CPD}}$  bit in the Configuration Word register (Register 12-1) to '0'.

When the data memory is code-protected, the CPU is able to read and write data to the data EEPROM. It is recommended to code-protect the program memory when code-protecting data memory. This prevents anyone from programming zeroes over the existing code (which will execute as NOPs) to reach an added routine, programmed in unused program memory, which outputs the contents of data memory. Programming unused locations in program memory to '0' will also help prevent data memory code protection from becoming breached.

**TABLE 10-1: SUMMARY OF ASSOCIATED DATA EEPROM REGISTERS**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
INTCON	GIE	PEIE	T0IE	INTE	GPIE	T0IF	INTF	GPIF	0000 0000	0000 0000
PIR1	EEIF	ADIF	CCP1IF	—	CMIF	OSFIF	TMR2IF	TMR1IF	000- 0000	000- 0000
PIE1	EEIE	ADIE	CCP1IE	—	CMIE	OSFIE	TMR2IE	TMR1IE	000- 0000	000- 0000
EEDAT	EEDAT7	EEDAT6	EEDAT5	EEDAT4	EEDAT3	EEDAT2	EEDAT1	EEDAT0	0000 0000	0000 0000
EEADR	EEADR7	EEADR6	EEADR5	EEADR4	EEADR3	EEADR2	EEADR1	EEADR0	0000 0000	0000 0000
EECON1	—	—	—	—	WRERR	WREN	WR	RD	---- x000	---- q000
EECON2 <sup>(1)</sup>	EEPROM Control Register 2								---- ----	---- ----

**Legend:** x = unknown, u = unchanged, — = unimplemented read as '0', q = value depends upon condition. Shaded cells are not used by the Data EEPROM module.

**Note 1:** EECON2 is not a physical register.

## 11.0 CAPTURE/COMPARE/PWM (CCP) MODULE

The Capture/Compare/PWM module is a peripheral which allows the user to time and control different events. In Capture mode, the peripheral allows the timing of the duration of an event. The Compare mode allows the user to trigger an external event when a predetermined amount of time has expired. The PWM mode can generate a Pulse-Width Modulated signal of varying frequency and duty cycle.

The timer resources used by the module are shown in Table 11-1

Additional information on CCP modules is available in the Application Note AN594, "Using the CCP Modules" (DS00594).

**TABLE 11-1: CCP MODE – TIMER RESOURCES REQUIRED**

CCP Mode	Timer Resource
Capture	Timer1
Compare	Timer1
PWM	Timer2

**REGISTER 11-1: CCP1CON: CCP1 CONTROL REGISTER**

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0
bit 7							bit 0

**Legend:**

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

bit 7-6 **Unimplemented:** Read as '0'

bit 5-4 **DC1B<1:0>:** PWM Duty Cycle Least Significant bits

Capture mode:

Unused.

Compare mode:

Unused.

PWM mode:

These bits are the two LSBs of the PWM duty cycle. The eight MSBs are found in CCPR1L.

bit 3-0 **CCP1M<3:0>:** CCP Mode Select bits

0000 = Capture/Compare/PWM off (resets CCP module)

0001 = Unused (reserved)

0010 = Unused (reserved)

0011 = Unused (reserved)

0100 = Capture mode, every falling edge

0101 = Capture mode, every rising edge

0110 = Capture mode, every 4th rising edge

0111 = Capture mode, every 16th rising edge

1000 = Compare mode, set output on match (CCP1IF bit is set)

1001 = Compare mode, clear output on match (CCP1IF bit is set)

1010 = Compare mode, generate software interrupt on match (CCP1IF bit is set, CCP1 pin is unaffected)

1011 = Compare mode, trigger special event (CCP1IF bit is set, TMR1 is reset and A/D conversion is started if the ADC module is enabled. CCP1 pin is unaffected.)

110x = PWM mode active-high

111x = PWM mode active-low

# PIC12F683

## 11.1 Capture Mode

In Capture mode, CCPR1H:CCPR1L captures the 16-bit value of the TMR1 register when an event occurs on pin CCP1. An event is defined as one of the following and is configured by the CCP1M<3:0> bits of the CCP1CON register:

- Every falling edge
- Every rising edge
- Every 4th rising edge
- Every 16th rising edge

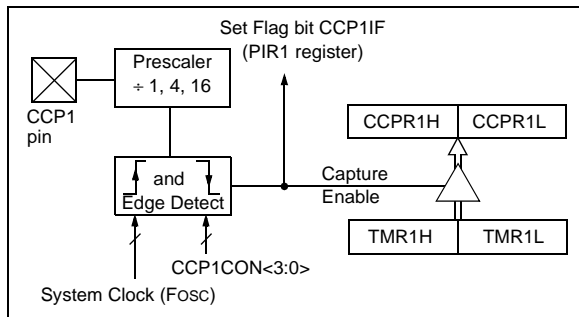
When a capture is made, the Interrupt Request Flag bit CCP1IF of the PIR1 register is set. The interrupt flag must be cleared in software. If another capture occurs before the value in the CCPR1H, CCPR1L register pair is read, the old captured value is overwritten by the new captured value (see Figure 11-1).

### 11.1.1 CCP1 PIN CONFIGURATION

In Capture mode, the CCP1 pin should be configured as an input by setting the associated TRIS control bit.

**Note:** If the CCP1 pin is configured as an output, a write to the GPIO port can cause a capture condition.

**FIGURE 11-1: CAPTURE MODE OPERATION BLOCK DIAGRAM**



### 11.1.2 TIMER1 MODE SELECTION

Timer1 must be running in Timer mode or Synchronized Counter mode for the CCP module to use the capture feature. In Asynchronous Counter mode, the capture operation may not work.

### 11.1.3 SOFTWARE INTERRUPT

When the Capture mode is changed, a false capture interrupt may be generated. The user should keep the CCP1IE interrupt enable bit of the PIE1 register clear to avoid false interrupts. Additionally, the user should clear the CCP1IF interrupt flag bit of the PIR1 register following any change in operating mode.

### 11.1.4 CCP PRESCALER

There are four prescaler settings specified by the CCP1M<3:0> bits of the CCP1CON register. Whenever the CCP module is turned off, or the CCP module is not in Capture mode, the prescaler counter is cleared. Any Reset will clear the prescaler counter.

Switching from one capture prescaler to another does not clear the prescaler and may generate a false interrupt. To avoid this unexpected operation, turn the module off by clearing the CCP1CON register before changing the prescaler (see Example 11-1).

**EXAMPLE 11-1: CHANGING BETWEEN CAPTURE PRESCALERS**

```

BANKSEL CCP1CON    ;Set Bank bits to point
                    ;to CCP1CON
CLRWF  CCP1CON      ;Turn CCP module off
MOVLW  NEW_CAPT_PS  ;Load the W reg with
                    ; the new prescaler
MOVWF  CCP1CON      ; move value and CCP ON
MOVWF  CCP1CON      ;Load CCP1CON with this
                    ; value
  
```



## 11.2 Compare Mode

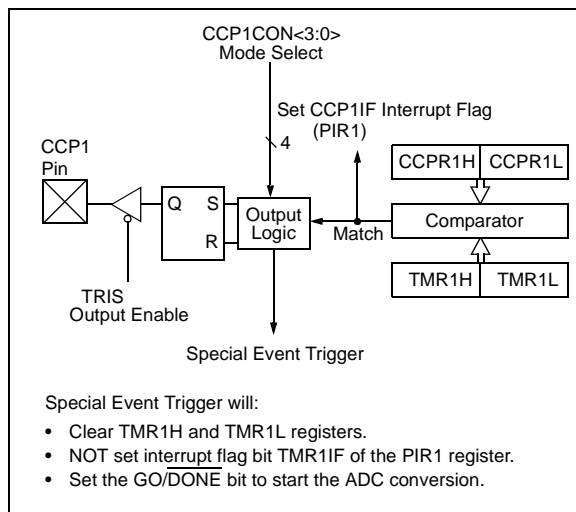
In Compare mode, the 16-bit CCPR1 register value is constantly compared against the TMR1 register pair value. When a match occurs, the CCP1 module may:

- Toggle the CCP1 output.
- Set the CCP1 output.
- Clear the CCP1 output.
- Generate a Special Event Trigger.
- Generate a Software Interrupt.

The action on the pin is based on the value of the CCP1M<3:0> control bits of the CCP1CON register.

All Compare modes can generate an interrupt.

**FIGURE 11-2: COMPARE MODE OPERATION BLOCK DIAGRAM**



### 11.2.1 CCP1 PIN CONFIGURATION

The user must configure the CCP1 pin as an output by clearing the associated TRIS bit.

**Note:** Clearing the CCP1CON register will force the CCP1 compare output latch to the default low level. This is not the GPIO I/O data latch.

### 11.2.2 TIMER1 MODE SELECTION

In Compare mode, Timer1 must be running in either Timer mode or Synchronized Counter mode. The compare operation may not work in Asynchronous Counter mode.

### 11.2.3 SOFTWARE INTERRUPT MODE

When Generate Software Interrupt mode is chosen (CCP1M<3:0> = 1010), the CCP1 module does not assert control of the CCP1 pin (see the CCP1CON register).

### 11.2.4 SPECIAL EVENT TRIGGER

When Special Event Trigger mode is chosen (CCP1M<3:0> = 1011), the CCP1 module does the following:

- Resets Timer1
- Starts an ADC conversion if ADC is enabled

The CCP1 module does not assert control of the CCP1 pin in this mode (see the CCP1CON register).

The Special Event Trigger output of the CCP occurs immediately upon a match between the TMR1H, TMR1L register pair and the CCPR1H, CCPR1L register pair. The TMR1H, TMR1L register pair is not reset until the next rising edge of the Timer1 clock. This allows the CCPR1H, CCPR1L register pair to effectively provide a 16-bit programmable period register for Timer1.

**Note 1:** The Special Event Trigger from the CCP module does not set interrupt flag bit TMRxIF of the PIR1 register.

- 2: Removing the match condition by changing the contents of the CCPR1H and CCPR1L register pair, between the clock edge that generates the Special Event Trigger and the clock edge that generates the Timer1 Reset, will preclude the Reset from occurring.

## 11.3 PWM Mode

The PWM mode generates a Pulse-Width Modulated signal on the CCP1 pin. The duty cycle, period and resolution are determined by the following registers:

- PR2
- T2CON
- CCPR1L
- CCP1CON

In Pulse-Width Modulation (PWM) mode, the CCP module produces up to a 10-bit resolution PWM output on the CCP1 pin. Since the CCP1 pin is multiplexed with the PORT data latch, the TRIS for that pin must be cleared to enable the CCP1 pin output driver.

**Note:** Clearing the CCP1CON register will relinquish CCP1 control of the CCP1 pin.

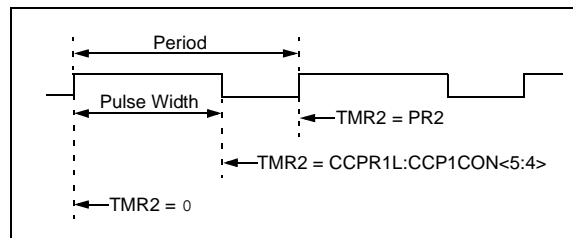
Figure 11-1 shows a simplified block diagram of PWM operation.

Figure 11-4 shows a typical waveform of the PWM signal.

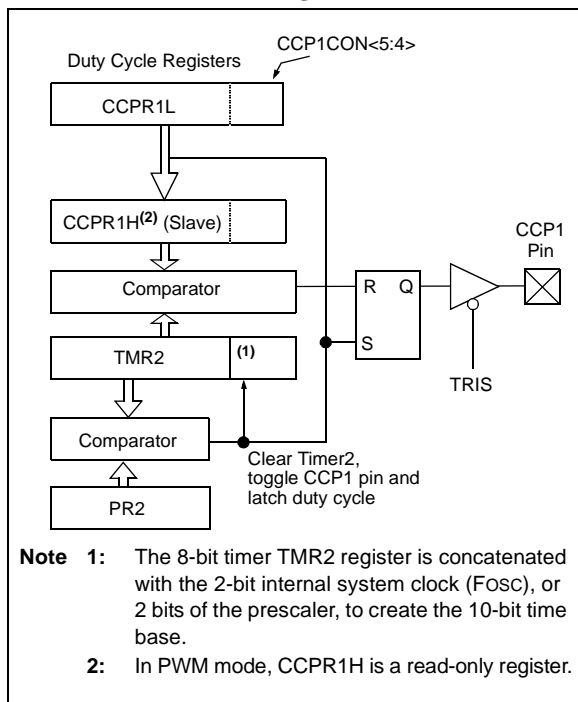
For a step-by-step procedure on how to set up the CCP module for PWM operation, see **Section 11.3.7 “Setup for PWM Operation”**.

The PWM output (Figure 11-4) has a time base (period) and a time that the output stays high (duty cycle).

**FIGURE 11-4: CCP PWM OUTPUT**



**FIGURE 11-3: SIMPLIFIED PWM BLOCK DIAGRAM**



## 11.3.1 PWM PERIOD

The PWM period is specified by the PR2 register of Timer2. The PWM period can be calculated using the formula of Equation 11-1.

### EQUATION 11-1: PWM PERIOD

$$PWM\ Period = [(PR2) + 1] \cdot 4 \cdot TOSC \cdot (TMR2\ Prescale\ Value)$$

When TMR2 is equal to PR2, the following three events occur on the next increment cycle:

- TMR2 is cleared
- The CCP1 pin is set. (Exception: If the PWM duty cycle = 0%, the pin will not be set.)
- The PWM duty cycle is latched from CCPR1L into CCPR1H.

**Note:** The Timer2 postscaler (see **Section 7.0 “Timer2 Module”**) is not used in the determination of the PWM frequency.

## 11.3.2 PWM DUTY CYCLE

The PWM duty cycle is specified by writing a 10-bit value to multiple registers: CCPR1L register and DC1B<1:0> bits of the CCP1CON register. The CCPR1L contains the eight MSBs and the CCP1<1:0> bits of the CCP1CON register contain the two LSbs. CCPR1L and DC1B<1:0> bits of the CCP1CON register can be written to at any time. The duty cycle value is not latched into CCPR1H until after the period completes (i.e., a match between PR2 and TMR2 registers occurs). While using the PWM, the CCPR1H register is read-only.

Equation 11-2 is used to calculate the PWM pulse width.

Equation 11-3 is used to calculate the PWM duty cycle ratio.

### EQUATION 11-2: PULSE WIDTH

$$Pulse\ Width = (CCPR1L:CCP1CON<5:4>) \cdot TOSC \cdot (TMR2\ Prescale\ Value)$$

### EQUATION 11-3: DUTY CYCLE RATIO

$$Duty\ Cycle\ Ratio = \frac{(CCPR1L:CCP1CON<5:4>)}{4(PR2 + 1)}$$

The CCPR1H register and a 2-bit internal latch are used to double buffer the PWM duty cycle. This double buffering is essential for glitchless PWM operation.

The 8-bit timer TMR2 register is concatenated with either the 2-bit internal system clock (Fosc), or 2 bits of the prescaler, to create the 10-bit time base. The system clock is used if the Timer2 prescaler is set to 1:1.

When the 10-bit time base matches the CCPR1H and 2-bit latch, then the CCP1 pin is cleared (see Figure 11-1).

## 11.3.3 PWM RESOLUTION

The resolution determines the number of available duty cycles for a given period. For example, a 10-bit resolution will result in 1024 discrete duty cycles, whereas an 8-bit resolution will result in 256 discrete duty cycles.

The maximum PWM resolution is 10 bits when PR2 is 255. The resolution is a function of the PR2 register value as shown by Equation 11-4.

### EQUATION 11-4: PWM RESOLUTION

$$Resolution = \frac{\log[4(PR2 + 1)]}{\log(2)}\ bits$$

**Note:** If the pulse width value is greater than the period the assigned PWM pin(s) will remain unchanged.

**TABLE 11-2: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS (Fosc = 20 MHz)**

PWM Frequency	1.22 kHz	4.88 kHz	19.53 kHz	78.12 kHz	156.3 kHz	208.3 kHz
Timer Prescale (1, 4, 16)	16	4	1	1	1	1
PR2 Value	0xFF	0xFF	0xFF	0x3F	0x1F	0x17
Maximum Resolution (bits)	10	10	10	8	7	6.6

**TABLE 11-3: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS (Fosc = 8 MHz)**

PWM Frequency	1.22 kHz	4.90 kHz	19.61 kHz	76.92 kHz	153.85 kHz	200.0 kHz
Timer Prescale (1, 4, 16)	16	4	1	1	1	1
PR2 Value	0x65	0x65	0x65	0x19	0x0C	0x09
Maximum Resolution (bits)	8	8	8	6	5	5

## 11.3.4 OPERATION IN SLEEP MODE

In Sleep mode, the TMR2 register will not increment and the state of the module will not change. If the CCP1 pin is driving a value, it will continue to drive that value. When the device wakes up, TMR2 will continue from its previous state.

## 11.3.5 CHANGES IN SYSTEM CLOCK FREQUENCY

The PWM frequency is derived from the system clock frequency. Any changes in the system clock frequency will result in changes to the PWM frequency. See **Section 3.0 “Oscillator Module (With Fail-Safe Clock Monitor)”** for additional details.

## 11.3.6 EFFECTS OF RESET

Any Reset will force all ports to Input mode and the CCP registers to their Reset states.

## 11.3.7 SETUP FOR PWM OPERATION

The following steps should be taken when configuring the CCP module for PWM operation:

1. Disable the PWM pin (CCP1) output drivers by setting the associated TRIS bit.
2. Set the PWM period by loading the PR2 register.
3. Configure the CCP module for the PWM mode by loading the CCP1CON register with the appropriate values.
4. Set the PWM duty cycle by loading the CCPR1L register and DC1B bits of the CCP1CON register.
5. Configure and start Timer2:
  - Clear the TMR2IF interrupt flag bit of the PIR1 register.
  - Set the Timer2 prescale value by loading the T2CKPS bits of the T2CON register.
  - Enable Timer2 by setting the TMR2ON bit of the T2CON register.
6. Enable PWM output after a new PWM cycle has started:
  - Wait until Timer2 overflows (TMR2IF bit of the PIR1 register is set).
  - Enable the CCP1 pin output driver by clearing the associated TRIS bit.

**TABLE 11-4: REGISTERS ASSOCIATED WITH CAPTURE, COMPARE AND TIMER1**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
CCP1CON	—	—	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	--00 0000
CCPR1L	Capture/Compare/PWM Register 1 Low Byte (LSB)								xxxx xxxx	xxxx xxxx
CCPR1H	Capture/Compare/PWM Register 1 High Byte (MSB)								xxxx xxxx	xxxx xxxx
CMCON1	—	—	—	—	—	—	T1GSS	CMSYNC	---- --10	---- --10
INTCON	GIE	PEIE	T0IE	INTE	GPIE	T0IF	INTF	GPIF	0000 0000	0000 000x
PIE1	EEIE	ADIE	CCP1IE	—	CMIE	OSFIE	TMR2IE	TMR1IE	000- 0000	000- 0000
PIR1	EEIF	ADIF	CCP1IF	—	CMIF	OSFIF	TMR2IF	TMR1IF	000- 0000	000- 0000
T1CON	T1GINV	TMR1GE	T1CKPS1	T1CKPS0	T1OSCEN	T1SYN $\bar{C}$	TMR1CS	TMR1ON	0000 0000	0000 0000
TMR1L	Holding Register for the Least Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	xxxx xxxx
TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	xxxx xxxx
TRISIO	—	—	TRISIO5	TRISIO4	TRISIO3	TRISIO2	TRISIO1	TRISIO0	--11 1111	--11 1111

**Legend:** — = Unimplemented locations, read as '0', u = unchanged, x = unknown. Shaded cells are not used by the Capture and Compare.

**TABLE 11-5: REGISTERS ASSOCIATED WITH PWM AND TIMER2**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
CCP1CON	—	—	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	--00 0000
CCPR1L	Capture/Compare/PWM Register 1 Low Byte (LSB)								xxxx xxxx	xxxx xxxx
CCPR1H	Capture/Compare/PWM Register 1 High Byte (MSB)								xxxx xxxx	xxxx xxxx
INTCON	GIE	PEIE	T0IE	INTE	GPIE	T0IF	INTF	GPIF	0000 0000	0000 000x
PIE1	EEIE	ADIE	CCP1IE	—	CMIE	OSFIE	TMR2IE	TMR1IE	000- 0000	-000 0000
PIR1	EEIF	ADIF	CCP1IF	—	CMIF	OSFIF	TMR2IF	TMR1IF	000- 0000	-000 0000
PR2	Timer2 Period Register								1111 1111	1111 1111
T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	-000 0000
TMR2	Timer2 Module Register								0000 0000	0000 0000
TRISIO	—	—	TRISIO5	TRISIO4	TRISIO3	TRISIO2	TRISIO1	TRISIO0	--11 1111	--11 1111

**Legend:** — = Unimplemented locations, read as '0', u = unchanged, x = unknown. Shaded cells are not used by the PWM.

# PIC12F683

---

NOTES:

## 12.0 SPECIAL FEATURES OF THE CPU

The PIC12F683 has a host of features intended to maximize system reliability, minimize cost through elimination of external components, provide power saving features and offer code protection.

These features are:

- Reset
  - Power-on Reset (POR)
  - Power-up Timer (PWRT)
  - Oscillator Start-up Timer (OST)
  - Brown-out Reset (BOR)
- Interrupts
- Watchdog Timer (WDT)
- Oscillator Selection
- Sleep
- Code Protection
- ID Locations
- In-Circuit Serial Programming™

The PIC12F683 has two timers that offer necessary delays on power-up. One is the Oscillator Start-up Timer (OST), intended to keep the chip in Reset until the crystal oscillator is stable. The other is the Power-up Timer (PWRT), which provides a fixed delay of 64 ms (nominal) on power-up only, designed to keep the part in Reset while the power supply stabilizes. There is also circuitry to reset the device if a brown-out occurs, which can use the Power-up Timer to provide at least a 64 ms Reset. With these three functions on-chip, most applications need no external Reset circuitry.

The Sleep mode is designed to offer a very low-current Power-down mode. The user can wake-up from Sleep through:

- External Reset
- Watchdog Timer Wake-up
- An interrupt

Several oscillator options are also made available to allow the part to fit the application. The INTOSC option saves system cost while the LP crystal option saves power. A set of Configuration bits are used to select various options (see Register 12-1).

<p><b>Note:</b> Address 2007h is beyond the user program memory space. It belongs to the special configuration memory space (2000h-3FFFh), which can be accessed only during programming. See “PIC12F6XX/16F6XX Memory Programming Specification” (DS41204) for more information.</p>
---

### 12.1 Configuration Bits

The Configuration bits can be programmed (read as ‘0’), or left unprogrammed (read as ‘1’) to select various device configurations as shown in Register 12-1. These bits are mapped in program memory location 2007h.

# PIC12F683

## REGISTER 12-1: CONFIG: CONFIGURATION WORD REGISTER

—	—	—	—	FCMEN	IESO	BOREN1	BOREN0
bit 15				bit 8			

$\overline{\text{CPD}}$	$\overline{\text{CP}}$	MCLRE	$\overline{\text{PWRTÉ}}$	WDTE	FOSC2	FOSC1	FOSC0
bit 7				bit 0			

### Legend:

R = Readable bit	W = Writable bit	P = Programmable'	U = Unimplemented bit, read as '0'
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-12 **Unimplemented:** Read as '1'

bit 11 **FCMEN:** Fail-Safe Clock Monitor Enabled bit  
1 = Fail-Safe Clock Monitor is enabled  
0 = Fail-Safe Clock Monitor is disabled

bit 10 **IESO:** Internal External Switchover bit  
1 = Internal External Switchover mode is enabled  
0 = Internal External Switchover mode is disabled

bit 9-8 **BOREN<1:0>:** Brown-out Reset Selection bits<sup>(1)</sup>  
11 = BOR enabled  
10 = BOR enabled during operation and disabled in Sleep  
01 = BOR controlled by SBOR bit of the PCON register  
00 = BOR disabled

bit 7  **$\overline{\text{CPD}}$ :** Data Code Protection bit<sup>(2)</sup>  
1 = Data memory code protection is disabled  
0 = Data memory code protection is enabled

bit 6  **$\overline{\text{CP}}$ :** Code Protection bit<sup>(3)</sup>  
1 = Program memory code protection is disabled  
0 = Program memory code protection is enabled

bit 5 **MCLRE:** GP3/ $\overline{\text{MCLR}}$  pin function select bit<sup>(4)</sup>  
1 = GP3/ $\overline{\text{MCLR}}$  pin function is  $\overline{\text{MCLR}}$   
0 = GP3/ $\overline{\text{MCLR}}$  pin function is digital input,  $\overline{\text{MCLR}}$  internally tied to VDD

bit 4  **$\overline{\text{PWRTÉ}}$ :** Power-up Timer Enable bit  
1 = PWRT disabled  
0 = PWRT enabled

bit 3 **WDTE:** Watchdog Timer Enable bit  
1 = WDT enabled  
0 = WDT disabled and can be enabled by SWDTEN bit of the WDTCON register

bit 2-0 **FOSC<2:0>:** Oscillator Selection bits  
111 = RC oscillator: CLKOUT function on GP4/OSC2/CLKOUT pin, RC on GP5/OSC1/CLKIN  
110 = RCIO oscillator: I/O function on GP4/OSC2/CLKOUT pin, RC on GP5/OSC1/CLKIN  
101 = INTOSC oscillator: CLKOUT function on GP4/OSC2/CLKOUT pin, I/O function on GP5/OSC1/CLKIN  
100 = INTOSCIO oscillator: I/O function on GP4/OSC2/CLKOUT pin, I/O function on GP5/OSC1/CLKIN  
011 = EC: I/O function on GP4/OSC2/CLKOUT pin, CLKIN on GP5/OSC1/CLKIN  
010 = HS oscillator: High-speed crystal/resonator on GP4/OSC2/CLKOUT and GP5/OSC1/CLKIN  
001 = XT oscillator: Crystal/resonator on GP4/OSC2/CLKOUT and GP5/OSC1/CLKIN  
000 = LP oscillator: Low-power crystal on GP4/OSC2/CLKOUT and GP5/OSC1/CLKIN

- Note** 1: Enabling Brown-out Reset does not automatically enable Power-up Timer.  
2: The entire data EEPROM will be erased when the code protection is turned off.  
3: The entire program memory will be erased when the code protection is turned off.  
4: When  $\overline{\text{MCLR}}$  is asserted in INTOSC or RC mode, the internal clock oscillator is disabled.



## 12.2 Calibration Bits

Brown-out Reset (BOR), Power-on Reset (POR) and 8 MHz internal oscillator (HFINTOSC) are factory calibrated. These calibration values are stored in fuses located in the Calibration Word (2009h). The Calibration Word is not erased when using the specified bulk erase sequence in the “PIC12F6XX/16F6XX Memory Programming Specification” (DS41244) and thus, does not require reprogramming.

## 12.3 Reset

The PIC12F683 differentiates between various kinds of Reset:

- Power-on Reset (POR)
- WDT Reset during normal operation
- WDT Reset during Sleep
- MCLR Reset during normal operation
- MCLR Reset during Sleep
- Brown-out Reset (BOR)

Some registers are not affected in any Reset condition; their status is unknown on POR and unchanged in any other Reset. Most other registers are reset to a “Reset state” on:

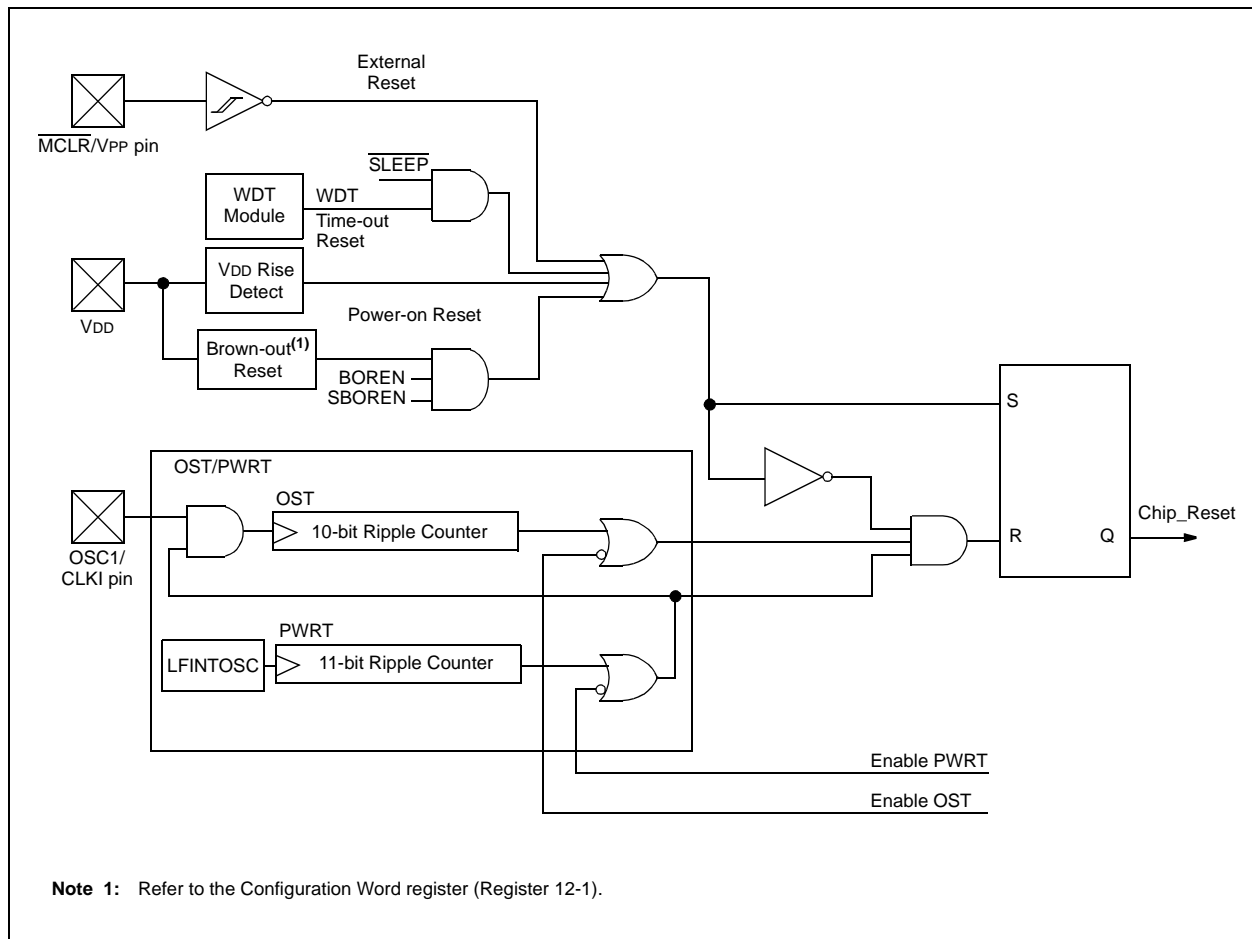
- Power-on Reset
- MCLR Reset
- MCLR Reset during Sleep
- WDT Reset
- Brown-out Reset (BOR)

WDT wake-up does not cause register resets in the same manner as a WDT Reset since wake-up is viewed as the resumption of normal operation.  $\overline{TO}$  and  $\overline{PD}$  bits are set or cleared differently in different Reset situations, as indicated in Table 12-2. Software can use these bits to determine the nature of the Reset. See Table 12-4 for a full description of Reset states of all registers.

A simplified block diagram of the On-Chip Reset Circuit is shown in Figure 12-1.

The MCLR Reset path has a noise filter to detect and ignore small pulses. See Section 15.0 “Electrical Specifications” for pulse-width specifications.

**FIGURE 12-1: SIMPLIFIED BLOCK DIAGRAM OF ON-CHIP RESET CIRCUIT**



# PIC12F683

## 12.3.1 POWER-ON RESET

The on-chip POR circuit holds the chip in Reset until VDD has reached a high enough level for proper operation. To take advantage of the POR, simply connect the  $\overline{\text{MCLR}}$  pin through a resistor to VDD. This will eliminate external RC components usually needed to create Power-on Reset. A maximum rise time for VDD is required. See **Section 15.0 “Electrical Specifications”** for details. If the BOR is enabled, the maximum rise time specification does not apply. The BOR circuitry will keep the device in Reset until VDD reaches VBOD (see **Section 12.3.4 “Brown-Out Reset (BOR)”**).

**Note:** The POR circuit does not produce an internal Reset when VDD declines. To re-enable the POR, VDD must reach VSS for a minimum of 100  $\mu\text{s}$ .

When the device starts normal operation (exits the Reset condition), device operating parameters (i.e., voltage, frequency, temperature, etc.) must be met to ensure operation. If these conditions are not met, the device must be held in Reset until the operating conditions are met.

For additional information, refer to the Application Note AN607, “Power-up Trouble Shooting” (DS00607).

## 12.3.2 $\overline{\text{MCLR}}$

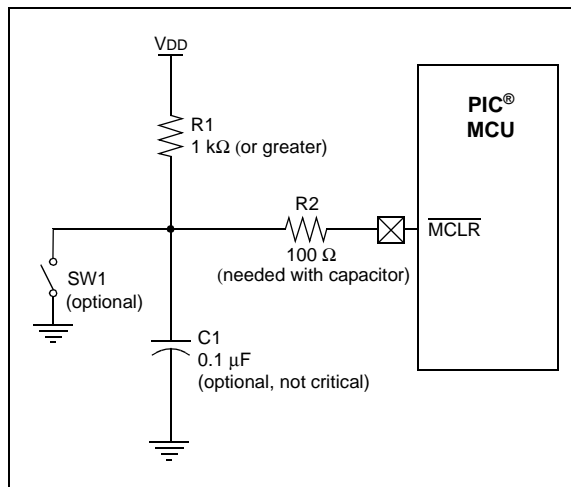
PIC12F683 has a noise filter in the  $\overline{\text{MCLR}}$  Reset path. The filter will detect and ignore small pulses.

It should be noted that a WDT Reset does not drive  $\overline{\text{MCLR}}$  pin low.

Voltages applied to the  $\overline{\text{MCLR}}$  pin that exceed its specification can result in both  $\overline{\text{MCLR}}$  Resets and excessive current beyond the device specification during the ESD event. For this reason, Microchip recommends that the  $\overline{\text{MCLR}}$  pin no longer be tied directly to VDD. The use of an RC network, as shown in Figure 12-2, is suggested.

An internal  $\overline{\text{MCLR}}$  option is enabled by clearing the  $\overline{\text{MCLR}}$  bit in the Configuration Word register. When  $\overline{\text{MCLR}} = 0$ , the Reset signal to the chip is generated internally. When the  $\overline{\text{MCLR}} = 1$ , the GP3/ $\overline{\text{MCLR}}$  pin becomes an external Reset input. In this mode, the GP3/ $\overline{\text{MCLR}}$  pin has a weak pull-up to VDD.

**FIGURE 12-2: RECOMMENDED  $\overline{\text{MCLR}}$  CIRCUIT**



## 12.3.3 POWER-UP TIMER (PWRT)

The Power-up Timer provides a fixed 64 ms (nominal) time-out on power-up only, from POR or Brown-out Reset. The Power-up Timer operates from the 31 kHz LFINTOSC oscillator. For more information, see **Section 3.5 “Internal Clock Modes”**. The chip is kept in Reset as long as PWRT is active. The PWRT delay allows the VDD to rise to an acceptable level. A Configuration bit, PWRT $\overline{\text{TE}}$ , can disable (if set) or enable (if cleared or programmed) the Power-up Timer. The Power-up Timer should be enabled when Brown-out Reset is enabled, although it is not required.

The Power-up Timer delay will vary from chip-to-chip due to:

- VDD variation
- Temperature variation
- Process variation

See DC parameters for details (**Section 15.0 “Electrical Specifications”**).

**Note:** Voltage spikes below VSS at the  $\overline{\text{MCLR}}$  pin, inducing currents greater than 80 mA, may cause latch-up. Thus, a series resistor of 50-100  $\Omega$  should be used when applying a “low” level to the  $\overline{\text{MCLR}}$  pin, rather than pulling this pin directly to VSS.

## 12.3.4 BROWN-OUT RESET (BOR)

The BOREN0 and BOREN1 bits in the Configuration Word register select one of four BOR modes. Two modes have been added to allow software or hardware control of the BOR enable. When BOREN<1:0> = 01, the SBOREN bit of the PCON register enables/disables the BOR, allowing it to be controlled in software. By selecting BOREN<1:0> = 10, the BOR is automatically disabled in Sleep to conserve power and enabled on wake-up. In this mode, the SBOREN bit is disabled. See Register 12-1 for the Configuration Word definition.

A brown-out occurs when VDD falls below VBOR for greater than parameter TBOR (see **Section 15.0 “Electrical Specifications”**). The brown-out condition will reset the device. This will occur regardless of VDD slew rate. A Brown-out Reset may not occur if VDD falls below VBOR for less than parameter TBOR.

On any Reset (Power-on, Brown-out Reset, Watchdog Timer, etc.), the chip will remain in Reset until VDD rises above VBOR (see Figure 12-3). If enabled, the Power-up Timer will be invoked by the Reset and keep the chip in Reset an additional 64 ms.

**Note:** The Power-up Timer is enabled by the PWRTE bit in the Configuration Word register.

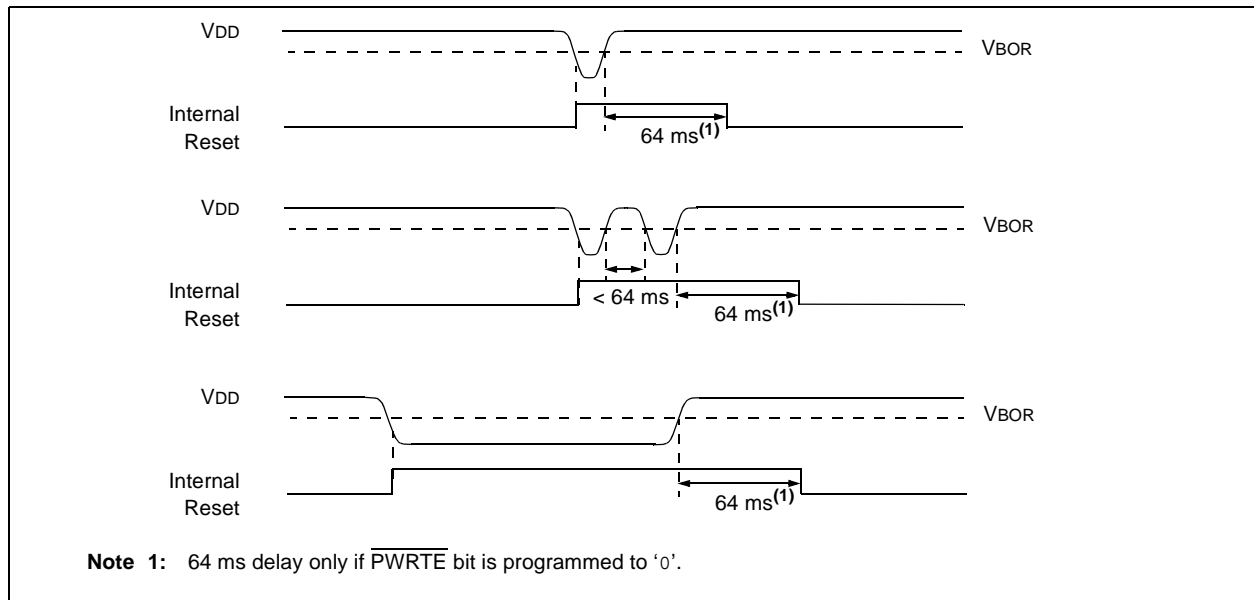
If VDD drops below VBOR while the Power-up Timer is running, the chip will go back into a Brown-out Reset and the Power-up Timer will be re-initialized. Once VDD rises above VBOR, the Power-up Timer will execute a 64 ms Reset.

## 12.3.5 BOR CALIBRATION

The PIC12F683 stores the BOR calibration values in fuses located in the Calibration Word register (2008h). The Calibration Word register is not erased when using the specified bulk erase sequence in the “PIC12F6XX/16F6XX Memory Programming Specification” (DS41204) and thus, does not require reprogramming.

**Note:** Address 2008h is beyond the user program memory space. It belongs to the special configuration memory space (2000h-3FFFh), which can be accessed only during programming. See “PIC12F6XX/16F6XX Memory Programming Specification” (DS41204) for more information.

**FIGURE 12-3: BROWN-OUT SITUATIONS**



# PIC12F683

## 12.3.6 TIME-OUT SEQUENCE

On power-up, the time-out sequence is as follows:

- PWRT time-out is invoked after POR has expired.
- OST is activated after the PWRT time-out has expired.

The total time-out will vary based on oscillator configuration and PWRTE bit status. For example, in EC mode with PWRTE bit erased (PWRT disabled), there will be no time-out at all. Figure 12-4, Figure 12-5 and Figure 12-6 depict time-out sequences. The device can execute code from the INTOSC while OST is active by enabling Two-Speed Start-up or Fail-Safe Monitor (see **Section 3.7.2 “Two-Speed Start-up Sequence”** and **Section 3.8 “Fail-Safe Clock Monitor”**).

Since the time-outs occur from the POR pulse, if  $\overline{\text{MCLR}}$  is kept low long enough, the time-outs will expire. Then, bringing  $\overline{\text{MCLR}}$  high will begin execution immediately (see Figure 12-5). This is useful for testing purposes or to synchronize more than one PIC12F683 device operating in parallel.

Table 12-5 shows the Reset conditions for some special registers, while Table 12-4 shows the Reset conditions for all the registers.

## 12.3.7 POWER CONTROL (PCON) REGISTER

The Power Control register PCON (address 8Eh) has two Status bits to indicate what type of Reset occurred last.

Bit 0 is  $\overline{\text{BOR}}$  (Brown-out).  $\overline{\text{BOR}}$  is unknown on Power-on Reset. It must then be set by the user and checked on subsequent Resets to see if  $\overline{\text{BOR}} = 0$ , indicating that a Brown-out has occurred. The  $\overline{\text{BOR}}$  Status bit is a “don’t care” and is not necessarily predictable if the brown-out circuit is disabled ( $\text{BOREN}<1:0> = 00$  in the Configuration Word register).

Bit 1 is  $\overline{\text{POR}}$  (Power-on Reset). It is a ‘0’ on Power-on Reset and unaffected otherwise. The user must write a ‘1’ to this bit following a Power-on Reset. On a subsequent Reset, if  $\overline{\text{POR}}$  is ‘0’, it will indicate that a Power-on Reset has occurred (i.e., VDD may have gone too low).

For more information, see **Section 4.2.4 “Ultra Low-Power Wake-up”** and **Section 12.3.4 “Brown-Out Reset (BOR)”**.

**TABLE 12-1: TIME-OUT IN VARIOUS SITUATIONS**

Oscillator Configuration	Power-up		Brown-out Reset		Wake-up from Sleep
	$\overline{\text{PWRTE}} = 0$	$\overline{\text{PWRTE}} = 1$	$\overline{\text{PWRTE}} = 0$	$\overline{\text{PWRTE}} = 1$	
XT, HS, LP	$\text{TPWRT} + 1024 \cdot \text{TOSC}$	$1024 \cdot \text{TOSC}$	$\text{TPWRT} + 1024 \cdot \text{TOSC}$	$1024 \cdot \text{TOSC}$	$1024 \cdot \text{TOSC}$
RC, EC, INTOSC	TPWRT	—	TPWRT	—	—

**TABLE 12-2: STATUS/PCON BITS AND THEIR SIGNIFICANCE**

$\overline{\text{POR}}$	$\overline{\text{BOR}}$	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Condition
0	x	1	1	Power-on Reset
u	0	1	1	Brown-out Reset
u	u	0	u	WDT Reset
u	u	0	0	WDT Wake-up
u	u	u	u	$\overline{\text{MCLR}}$ Reset during normal operation
u	u	1	0	$\overline{\text{MCLR}}$ Reset during Sleep

**Legend:** u = unchanged, x = unknown

**TABLE 12-3: SUMMARY OF REGISTERS ASSOCIATED WITH BROWN-OUT RESET**

Name	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets <sup>(1)</sup>
CONFIG <sup>(2)</sup>	BOREN1	BOREN0	$\overline{\text{CPD}}$	$\overline{\text{CP}}$	MCLRE	$\overline{\text{PWRTE}}$	WDTE	FOSC2	FOSC1	FOSC0	—	—
PCON			—	—	ULPWUE	SBOREN	—	—	$\overline{\text{POR}}$	$\overline{\text{BOR}}$	--01 --qq	--0u --uu
STATUS			IRP	RP1	RP0	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C	0001 1xxx	000q quuu

**Legend:** u = unchanged, x = unknown, — = unimplemented bit, reads as ‘0’, q = value depends on condition. Shaded cells are not used by BOR.  
**Note 1:** Other (non Power-up) Resets include  $\overline{\text{MCLR}}$  Reset and Watchdog Timer Reset during normal operation.  
**Note 2:** See Configuration Word register (Register 12-1) for operation of all register bits.

FIGURE 12-4: TIME-OUT SEQUENCE ON POWER-UP (DELAYED  $\overline{\text{MCLR}}$ )

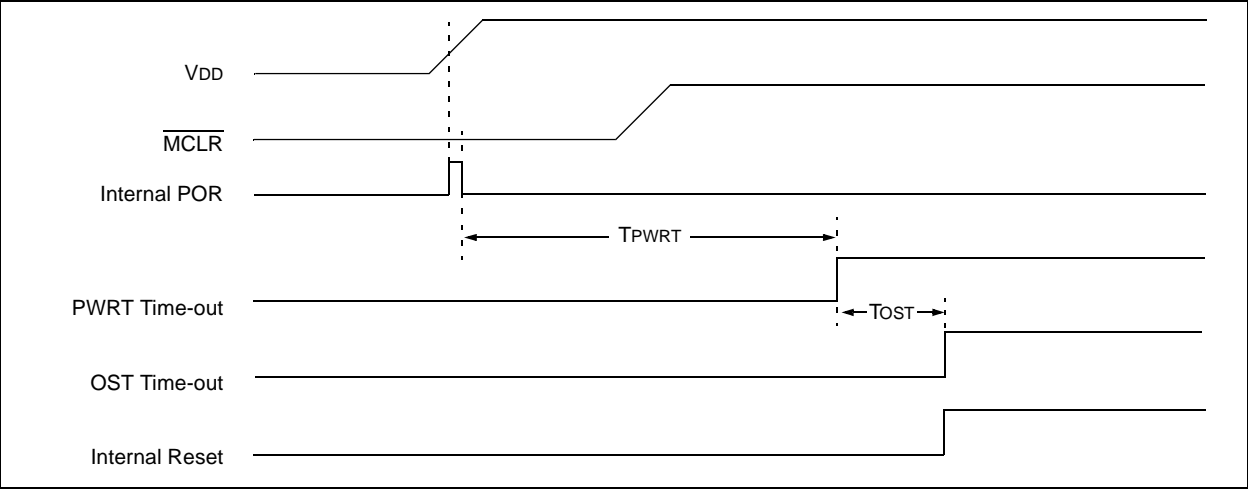


FIGURE 12-5: TIME-OUT SEQUENCE ON POWER-UP (DELAYED  $\overline{\text{MCLR}}$ )

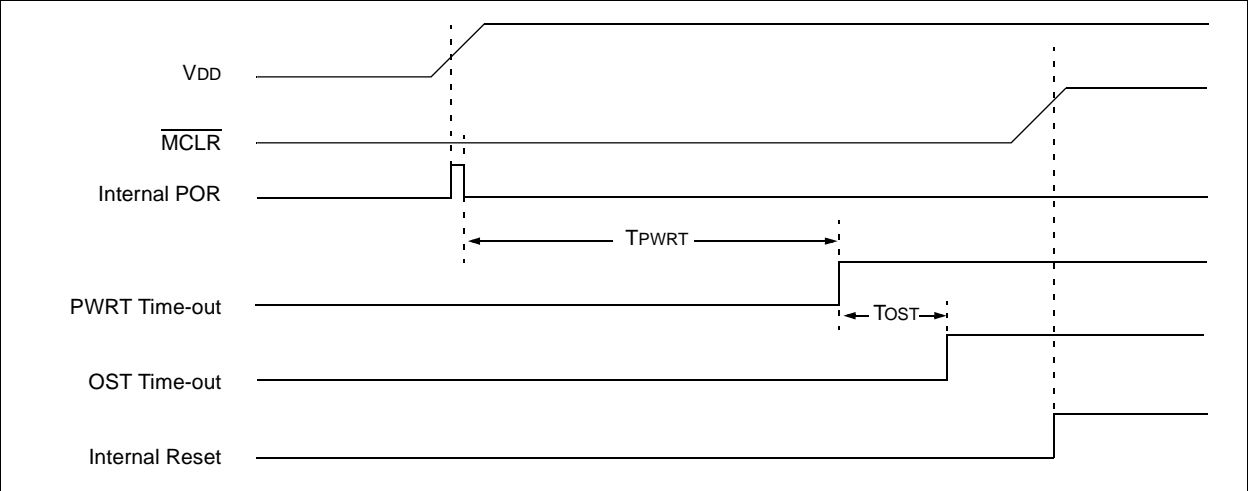
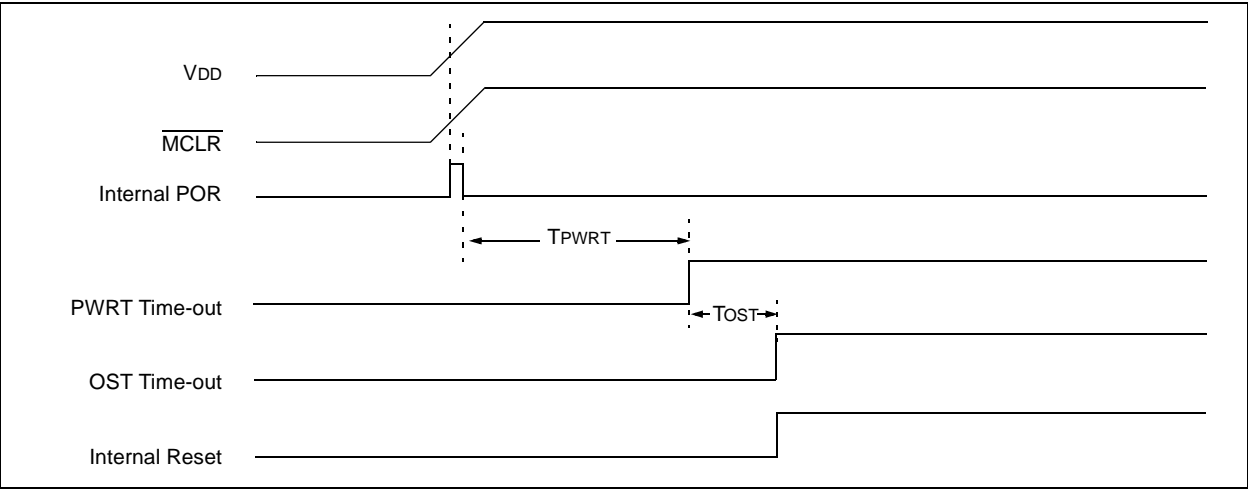


FIGURE 12-6: TIME-OUT SEQUENCE ON POWER-UP ( $\overline{\text{MCLR}}$  WITH  $V_{DD}$ )



# PIC12F683

**TABLE 12-4: INITIALIZATION CONDITION FOR REGISTERS**

Register	Address	Power-on Reset	MCLR Reset WDT Reset Brown-out Reset <sup>(1)</sup>	Wake-up from Sleep through Interrupt Wake-up from Sleep through WDT Time-out
W	—	xxxx xxxx	uuuu uuuu	uuuu uuuu
INDF	00h/80h	xxxx xxxx	xxxx xxxx	uuuu uuuu
TMR0	01h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PCL	02h/82h	0000 0000	0000 0000	PC + 1 <sup>(3)</sup>
STATUS	03h/83h	0001 1xxx	000q quuu <sup>(4)</sup>	uuuq quuu <sup>(4)</sup>
FSR	04h/84h	xxxx xxxx	uuuu uuuu	uuuu uuuu
GPIO	05h	--x0 x000	--x0 x000	--uu uuuu
PCLATH	0Ah/8Ah	---0 0000	---0 0000	---u uuuu
INTCON	0Bh/8Bh	0000 0000	0000 0000	uuuu uuuu <sup>(2)</sup>
PIR1	0Ch	0000 0000	0000 0000	uuuu uuuu <sup>(2)</sup>
TMR1L	0Eh	xxxx xxxx	uuuu uuuu	uuuu uuuu
TMR1H	0Fh	xxxx xxxx	uuuu uuuu	uuuu uuuu
T1CON	10h	0000 0000	uuuu uuuu	-uuu uuuu
TMR2	11h	0000 0000	0000 0000	uuuu uuuu
T2CON	12h	-000 0000	-000 0000	-uuu uuuu
CCPR1L	13h	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCPR1H	14h	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCP1CON	15h	--00 0000	--00 0000	--uu uuuu
WDTCON	18h	---0 1000	---0 1000	---u uuuu
CMCON0	19h	0000 0000	0000 0000	uuuu uuuu
CMCON1	20h	---- --10	---- --10	---- --uu
ADRESH	1Eh	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADCON0	1Fh	00-- 0000	00-- 0000	uu-- uuuu
OPTION_REG	81h	1111 1111	1111 1111	uuuu uuuu
TRISIO	85h	--11 1111	--11 1111	--uu uuuu
PIE1	8Ch	0000 0000	0000 0000	uuuu uuuu
PCON	8Eh	--01 --0x	--0u --uu <sup>(1,5)</sup>	--uu --uu
OSCCON	8Fh	-110 q000	-110 q000	-uuu uuuu
OSCTUNE	90h	---0 0000	---u uuuu	---u uuuu
PR2	92h	1111 1111	1111 1111	1111 1111
WPU	95h	--11 -111	--11 -111	uuuu uuuu
IOC	96h	--00 0000	--00 0000	--uu uuuu
VRCON	99h	0-0- 0000	0-0- 0000	u-u- uuuu
EEDAT	9Ah	0000 0000	0000 0000	uuuu uuuu
EEADR	9Bh	0000 0000	0000 0000	uuuu uuuu

**Legend:** u = unchanged, x = unknown, – = unimplemented bit, reads as '0', q = value depends on condition.

**Note 1:** If VDD goes too low, Power-on Reset will be activated and registers will be affected differently.

**2:** One or more bits in INTCON and/or PIR1 will be affected (to cause wake-up).

**3:** When the wake-up is due to an interrupt and the GIE bit is set, the PC is loaded with the interrupt vector (0004h).

**4:** See Table 12-5 for Reset value for specific condition.

**5:** If Reset was due to brown-out, then bit 0 = 0. All other Resets will cause bit 0 = u.

**TABLE 12-4: INITIALIZATION CONDITION FOR REGISTERS (CONTINUED)**

Register	Address	Power-on Reset	MCLR Reset WDT Reset Brown-out Reset <sup>(1)</sup>	Wake-up from Sleep through Interrupt Wake-up from Sleep through WDT Time-out
EECON1	9Ch	---- x000	---- q000	---- uuuu
EECON2	9Dh	---- ----	---- ----	---- ----
ADRESL	9Eh	xxxx xxxx	uuuu uuuu	uuuu uuuu
ANSEL	9Fh	-000 1111	-000 1111	-uuu uuuu

**Legend:** u = unchanged, x = unknown, – = unimplemented bit, reads as '0', q = value depends on condition.

**Note 1:** If VDD goes too low, Power-on Reset will be activated and registers will be affected differently.

2: One or more bits in INTCON and/or PIR1 will be affected (to cause wake-up).

3: When the wake-up is due to an interrupt and the GIE bit is set, the PC is loaded with the interrupt vector (0004h).

4: See Table 12-5 for Reset value for specific condition.

5: If Reset was due to brown-out, then bit 0 = 0. All other Resets will cause bit 0 = u.

**TABLE 12-5: INITIALIZATION CONDITION FOR SPECIAL REGISTERS**

Condition	Program Counter	Status Register	PCON Register
Power-on Reset	000h	0001 1xxx	--01 --0x
MCLR Reset during Normal Operation	000h	000u uuuu	--0u --uu
MCLR Reset during Sleep	000h	0001 0uuu	--0u --uu
WDT Reset	000h	0000 uuuu	--0u --uu
WDT Wake-up	PC + 1	uuu0 0uuu	--uu --uu
Brown-out Reset	000h	0001 1uuu	--01 --10
Interrupt Wake-up from Sleep	PC + 1 <sup>(1)</sup>	uuu1 0uuu	--uu --uu

**Legend:** u = unchanged, x = unknown, – = unimplemented bit, reads as '0'.

**Note 1:** When the wake-up is due to an interrupt and Global Interrupt Enable bit, GIE, is set, the PC is loaded with the interrupt vector (0004h) after execution of PC + 1.

## 12.4 Interrupts

The PIC12F683 has multiple interrupt sources:

- External Interrupt GP2/INT
- Timer0 Overflow Interrupt
- GPIO Change Interrupts
- Comparator Interrupt
- A/D Interrupt
- Timer1 Overflow Interrupt
- Timer2 Match Interrupt
- EEPROM Data Write Interrupt
- Fail-Safe Clock Monitor Interrupt
- CCP Interrupt

The Interrupt Control register (INTCON) and Peripheral Interrupt Request Register 1 (PIR1) record individual interrupt requests in flag bits. The INTCON register also has individual and global interrupt enable bits.

The Global Interrupt Enable bit, GIE of the INTCON register, enables (if set) all unmasked interrupts, or disables (if cleared) all interrupts. Individual interrupts can be disabled through their corresponding enable bits in the INTCON register and PIE1 register. GIE is cleared on Reset.

When an interrupt is serviced, the following actions occur automatically:

- The GIE is cleared to disable any further interrupt.
- The return address is pushed onto the stack.
- The PC is loaded with 0004h.

The Return from Interrupt instruction, `RETFIE`, exits the interrupt routine, as well as sets the GIE bit, which re-enables unmasked interrupts.

The following interrupt flags are contained in the INTCON register:

- INT Pin Interrupt
- GPIO Change Interrupt
- Timer0 Overflow Interrupt

The peripheral interrupt flags are contained in the PIR1 register. The corresponding interrupt enable bit is contained in the PIE1 register.

The following interrupt flags are contained in the PIR1 register:

- EEPROM Data Write Interrupt
- A/D Interrupt
- Comparator Interrupt
- Timer1 Overflow Interrupt
- Timer2 Match Interrupt
- Fail-Safe Clock Monitor Interrupt
- CCP Interrupt

For external interrupt events, such as the INT pin or GPIO change interrupt, the interrupt latency will be three or four instruction cycles. The exact latency depends upon when the interrupt event occurs (see Figure 12-8). The latency is the same for one or two-cycle instructions. Once in the Interrupt Service Routine, the source(s) of the interrupt can be determined by polling the interrupt flag bits. The interrupt flag bit(s) must be cleared in software before re-enabling interrupts to avoid multiple interrupt requests.

**Note 1:** Individual interrupt flag bits are set, regardless of the status of their corresponding mask bit or the GIE bit.

**2:** When an instruction that clears the GIE bit is executed, any interrupts that were pending for execution in the next cycle are ignored. The interrupts, which were ignored, are still pending to be serviced when the GIE bit is set again.

For additional information on Timer1, Timer2, comparators, ADC, data EEPROM or Enhanced CCP modules, refer to the respective peripheral section.

### 12.4.1 GP2/INT INTERRUPT

The external interrupt on the GP2/INT pin is edge-triggered; either on the rising edge if the INTEDG bit of the OPTION register is set, or the falling edge, if the INTEDG bit is clear. When a valid edge appears on the GP2/INT pin, the INTF bit of the INTCON register is set. This interrupt can be disabled by clearing the INTE control bit of the INTCON register. The INTF bit must be cleared by software in the Interrupt Service Routine before re-enabling this interrupt. The GP2/INT interrupt can wake-up the processor from Sleep, if the INTE bit was set prior to going into Sleep. See **Section 12.7 “Power-Down Mode (Sleep)”** for details on Sleep and Figure 12-10 for timing of wake-up from Sleep through GP2/INT interrupt.

**Note:** The ANSEL and CMCON0 registers must be initialized to configure an analog channel as a digital input. Pins configured as analog inputs will read ‘0’ and cannot generate an interrupt.



## 12.4.2 TIMER0 INTERRUPT

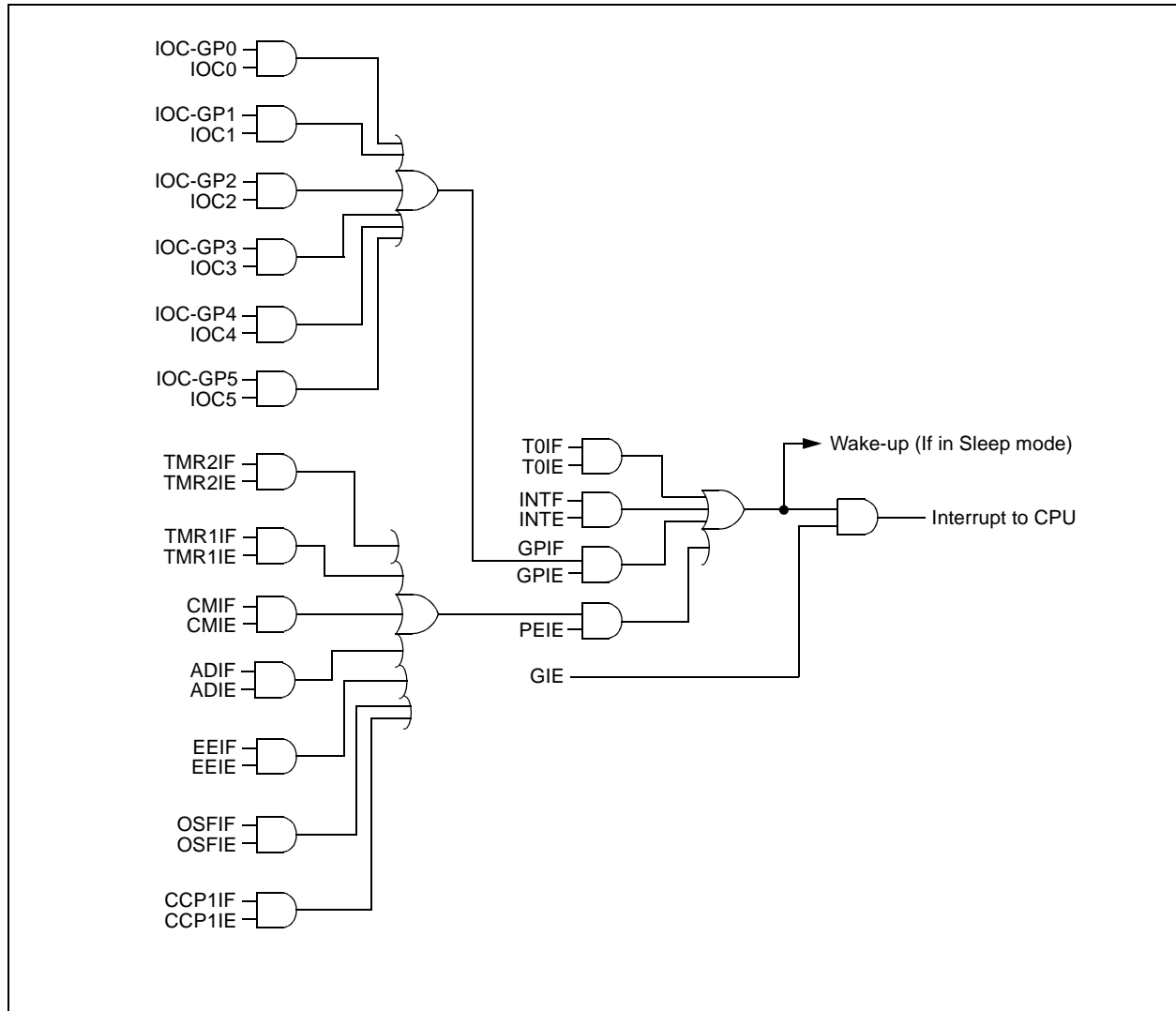
An overflow (FFh → 00h) in the TMR0 register will set the T0IF (INTCON<2>) bit. The interrupt can be enabled/disabled by setting/clearing the T0IE bit of the INTCON register. See **Section 5.0 “Timer0 Module”** for operation of the Timer0 module.

## 12.4.3 GPIO INTERRUPT

An input change on GPIO change sets the GPIF bit of the INTCON register. The interrupt can be enabled/disabled by setting/clearing the GPIE bit of the INTCON register. Plus, individual pins can be configured through the IOC register.

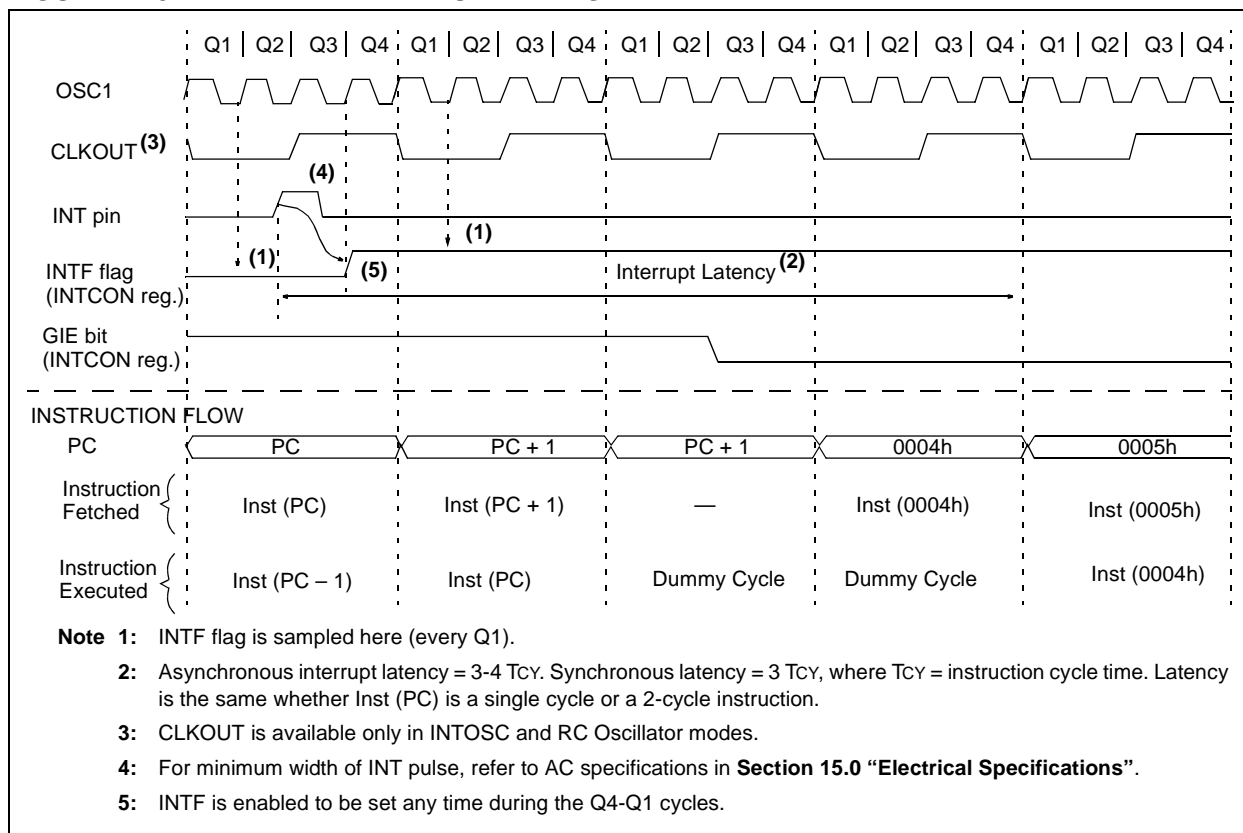
**Note:** If a change on the I/O pin should occur when any GPIO operation is being executed, then the GPIF interrupt flag may not get set.

**FIGURE 12-7: INTERRUPT LOGIC**



# PIC12F683

**FIGURE 12-8: INT PIN INTERRUPT TIMING**



**TABLE 12-6: SUMMARY OF REGISTERS ASSOCIATED WITH INTERRUPTS**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
INTCON	GIE	PEIE	T0IE	INTE	GPIE	T0IF	INTF	GPIF	0000 0000	0000 0000
IOC	—	—	IOC5	IOC4	IOC3	IOC2	IOC1	IOC0	--00 0000	--00 0000
PIR1	EEIF	ADIF	CCP1IF	—	CMIF	OSFIF	TMR2IF	TMR1IF	000- 0000	000- 0000
PIE1	EEIE	ADIE	CCP1IE	—	CMIE	OSFIE	TMR2IE	TMR1IE	000- 0000	000- 0000

**Legend:** x = unknown, u = unchanged, — = unimplemented read as ‘0’, q = value depends upon condition.  
 Shaded cells are not used by the interrupt module.

## 12.5 Context Saving During Interrupts

During an interrupt, only the return PC value is saved on the stack. Typically, users may wish to save key registers during an interrupt (e.g., W and STATUS registers). This must be implemented in software.

Since the lower 16 bytes of all banks are common in the PIC12F683 (see Figure 2-2), temporary holding registers, W\_TEMP and STATUS\_TEMP, should be placed in here. These 16 locations do not require banking and therefore, makes it easier to context save and restore. The same code shown in Example 12-1 can be used to:

- Store the W register.
- Store the STATUS register.
- Execute the ISR code.
- Restore the Status (and Bank Select Bit register).
- Restore the W register.

**Note:** The PIC12F683 normally does not require saving the PCLATH. However, if computed GOTO's are used in the ISR and the main code, the PCLATH must be saved and restored in the ISR.

### EXAMPLE 12-1: SAVING STATUS AND W REGISTERS IN RAM

```
MOVWF  W_TEMP          ;Copy W to TEMP register
SWAPF  STATUS,W        ;Swap status to be saved into W
                        ;Swaps are used because they do not affect the status bits
MOVWF  STATUS_TEMP     ;Save status to bank zero STATUS_TEMP register
:
: (ISR)                 ;Insert user code here
:
SWAPF  STATUS_TEMP,W   ;Swap STATUS_TEMP register into W
                        ;(sets bank to original state)
MOVWF  STATUS          ;Move W into STATUS register
SWAPF  W_TEMP,F        ;Swap W_TEMP
SWAPF  W_TEMP,W        ;Swap W_TEMP into W
```

# PIC12F683

## 12.6 Watchdog Timer (WDT)

The WDT has the following features:

- Operates from the LFINTOSC (31 kHz)
- Contains a 16-bit prescaler
- Shares an 8-bit prescaler with Timer0
- Time-out period is from 1 ms to 268 seconds
- Configuration bit and software controlled

WDT is cleared under certain conditions described in Table 12-7.

### 12.6.1 WDT OSCILLATOR

The WDT derives its time base from the 31 kHz LFINTOSC. The LTS bit of the OSCCON register does not reflect that the LFINTOSC is enabled.

The value of WDTCON is '---0 1000' on all Resets. This gives a nominal time base of 17 ms.

**Note:** When the Oscillator Start-up Timer (OST) is invoked, the WDT is held in Reset, because the WDT Ripple Counter is used by the OST to perform the oscillator delay count. When the OST count has expired, the WDT will begin counting (if enabled).

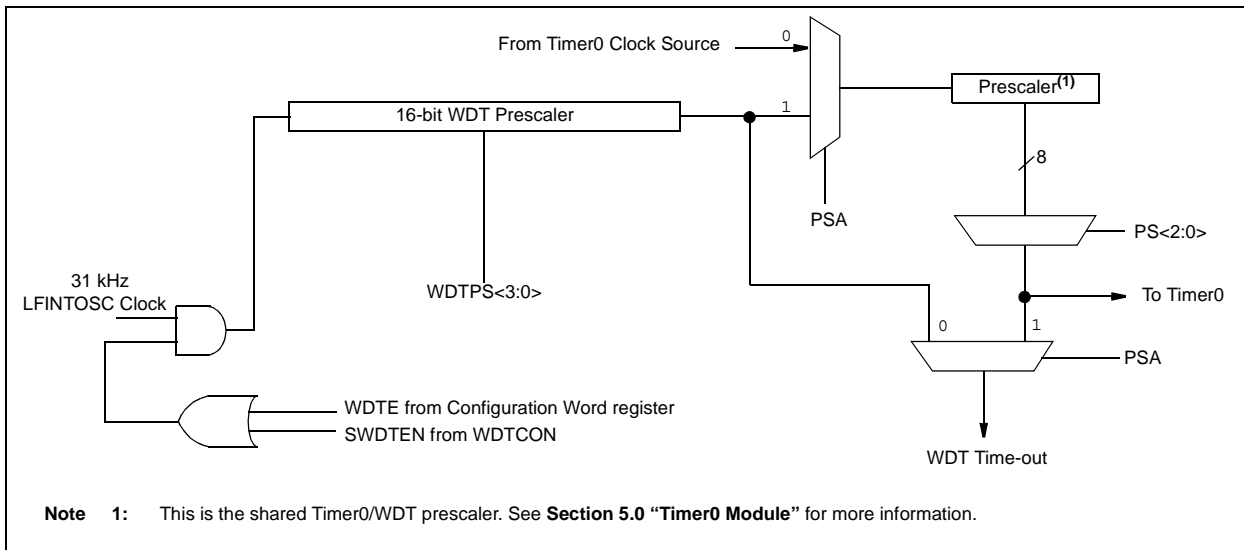
### 12.6.2 WDT CONTROL

The WDTE bit is located in the Configuration Word register. When set, the WDT runs continuously.

When the WDTE bit in the Configuration Word register is set, the SWDTEN bit of the WDTCON register has no effect. If WDTE is clear, then the SWDTEN bit can be used to enable and disable the WDT. Setting the bit will enable it and clearing the bit will disable it.

The PSA and PS<2:0> bits of the OPTION register have the same function as in previous versions of the PIC12F683 Family of microcontrollers. See **Section 5.0 "Timer0 Module"** for more information.

**FIGURE 12-9: WATCHDOG TIMER BLOCK DIAGRAM**



**TABLE 12-7: WDT STATUS**

Conditions	WDT
WDTE = 0	Cleared
CLRWDT Command	
Oscillator Fail Detected	
Exit Sleep + System Clock = T1OSC, EXTRC, INTRC, EXTCLK	
Exit Sleep + System Clock = XT, HS, LP	Cleared until the end of OST

## REGISTER 12-2: WDTCON: WATCHDOG TIMER CONTROL REGISTER

U-0	U-0	U-0	R/W-0	R/W-1	R/W-0	R/W-0	R/W-0
—	—	—	WDTPS3	WDTPS2	WDTPS1	WDTPS0	SWDTEN
bit 7							bit 0

### Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

bit 7-5 **Unimplemented:** Read as '0'

bit 4-1 **WDTPS<3:0>:** Watchdog Timer Period Select bits

Bit Value = Prescale Rate

0000 = 1:32

0001 = 1:64

0010 = 1:128

0011 = 1:256

0100 = 1:512 (Reset value)

0101 = 1:1024

0110 = 1:2048

0111 = 1:4096

1000 = 1:8192

1001 = 1:16384

1010 = 1:32768

1011 = 1:65536

1100 = Reserved

1101 = Reserved

1110 = Reserved

1111 = Reserved

bit 0 **SWDTEN:** Software Enable or Disable the Watchdog Timer<sup>(1)</sup>

1 = WDT is turned on

0 = WDT is turned off (Reset value)

**Note 1:** If WDTE Configuration bit = 1, then WDT is always enabled, irrespective of this control bit. If WDTE Configuration bit = 0, then it is possible to turn WDT on/off with this control bit.

**TABLE 12-8: SUMMARY OF REGISTERS ASSOCIATED WITH WATCHDOG TIMER**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
WDTCON	—	—	—	WDTPS3	WDTPS2	WDTPS1	WDTPS0	SWDTEN	---0 1000	---0 1000
OPTION_REG	GPPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
CONFIG	CPD	CP	MCLRE	PWRT	WDTE	FOSC2	FOSC1	FOSC0	—	—

**Legend:** Shaded cells are not used by the Watchdog Timer.

**Note 1:** See Register 12-1 for operation of all Configuration Word register bits.

## 12.7 Power-Down Mode (Sleep)

The Power-down mode is entered by executing a `SLEEP` instruction.

If the Watchdog Timer is enabled:

- WDT will be cleared but keeps running.
- $\overline{\text{PD}}$  bit in the STATUS register is cleared.
- $\overline{\text{TO}}$  bit is set.
- Oscillator driver is turned off.
- I/O ports maintain the status they had before `SLEEP` was executed (driving high, low or high-impedance).

For lowest current consumption in this mode, all I/O pins should be either at  $V_{DD}$  or  $V_{SS}$ , with no external circuitry drawing current from the I/O pin and the comparators and  $\text{CVREF}$  should be disabled. I/O pins that are high-impedance inputs should be pulled high or low externally to avoid switching currents caused by floating inputs. The  $\text{TOCKI}$  input should also be at  $V_{DD}$  or  $V_{SS}$  for lowest current consumption. The contribution from on-chip pull-ups on GPIO should be considered.

The  $\overline{\text{MCLR}}$  pin must be at a logic high level.

<b>Note:</b>	It should be noted that a Reset generated by a WDT time-out does not drive $\overline{\text{MCLR}}$ pin low.
--------------	--

### 12.7.1 WAKE-UP FROM SLEEP

The device can wake-up from Sleep through one of the following events:

1. External Reset input on  $\overline{\text{MCLR}}$  pin.
2. Watchdog Timer wake-up (if WDT was enabled).
3. Interrupt from  $\text{GP2/INT}$  pin, GPIO change or a peripheral interrupt.

The first event will cause a device Reset. The two latter events are considered a continuation of program execution. The  $\overline{\text{TO}}$  and  $\overline{\text{PD}}$  bits in the STATUS register can be used to determine the cause of a device Reset. The  $\overline{\text{PD}}$  bit, which is set on power-up, is cleared when Sleep is invoked.  $\overline{\text{TO}}$  bit is cleared if WDT wake-up occurred.

The following peripheral interrupts can wake the device from Sleep:

1. Timer1 interrupt. Timer1 must be operating as an asynchronous counter.
2. ECCP Capture mode interrupt.
3. A/D conversion (when A/D clock source is FRC).
4. EEPROM write operation completion.
5. Comparator output changes state.
6. Interrupt-on-change.
7. External Interrupt from INT pin.

Other peripherals cannot generate interrupts since during Sleep, no on-chip clocks are present.

When the `SLEEP` instruction is being executed, the next instruction ( $\text{PC} + 1$ ) is prefetched. For the device to wake-up through an interrupt event, the corresponding interrupt enable bit must be set (enabled). Wake-up occurs regardless of the state of the GIE bit. If the GIE bit is clear (disabled), the device continues execution at the instruction after the `SLEEP` instruction. If the GIE bit is set (enabled), the device executes the instruction after the `SLEEP` instruction, then branches to the interrupt address (0004h). In cases where the execution of the instruction following `SLEEP` is not desirable, the user should have a `NOP` after the `SLEEP` instruction.

<b>Note:</b>	If the global interrupts are disabled (GIE is cleared) and any interrupt source has both its interrupt enable bit and the corresponding interrupt flag bits set, the device will immediately wake-up from Sleep.
--------------	--

The WDT is cleared when the device wakes up from Sleep, regardless of the source of wake-up.

### 12.7.2 WAKE-UP USING INTERRUPTS

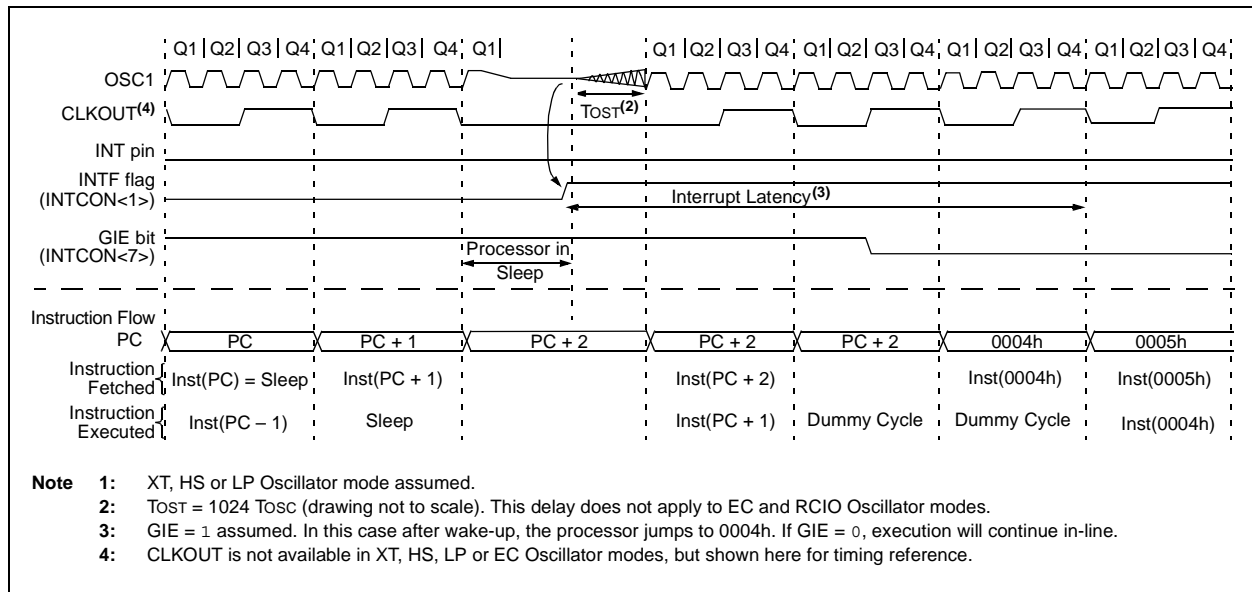
When global interrupts are disabled (GIE cleared) and any interrupt source has both its interrupt enable bit and interrupt flag bit set, one of the following will occur:

- If the interrupt occurs **before** the execution of a `SLEEP` instruction, the `SLEEP` instruction will complete as a `NOP`. Therefore, the WDT and WDT prescaler and postscaler (if enabled) will not be cleared, the  $\overline{\text{TO}}$  bit will not be set and the  $\overline{\text{PD}}$  bit will not be cleared.
- If the interrupt occurs **during or after** the execution of a `SLEEP` instruction, the device will immediately wake-up from Sleep. The `SLEEP` instruction is executed. Therefore, the WDT and WDT prescaler and postscaler (if enabled) will be cleared, the  $\overline{\text{TO}}$  bit will be set and the  $\overline{\text{PD}}$  bit will be cleared.

Even if the flag bits were checked before executing a `SLEEP` instruction, it may be possible for flag bits to become set before the `SLEEP` instruction completes. To determine whether a `SLEEP` instruction executed, test the  $\overline{\text{PD}}$  bit. If the  $\overline{\text{PD}}$  bit is set, the `SLEEP` instruction was executed as a `NOP`.

To ensure that the WDT is cleared, a `CLRWDT` instruction should be executed before a `SLEEP` instruction. See Figure 12-10 for more details.

**FIGURE 12-10: WAKE-UP FROM SLEEP THROUGH INTERRUPT**



## 12.8 Code Protection

If the code protection bit(s) have not been programmed, the on-chip program memory can be read out using ICSP™ for verification purposes.

**Note:** The entire data EEPROM and Flash program memory will be erased when the code protection is turned off. See the "PIC12F6XX/16F6XX Memory Programming Specification" (DS41204) for more information.

## 12.9 ID Locations

Four memory locations (2000h-2003h) are designated as ID locations where the user can store checksum or other code identification numbers. These locations are not accessible during normal execution, but are readable and writable during Program/Verify mode. Only the Least Significant 7 bits of the ID locations are used.

# PIC12F683

## 12.10 In-Circuit Serial Programming™

The PIC12F683 microcontrollers can be serially programmed while in the end application circuit. This is simply done with five connections for:

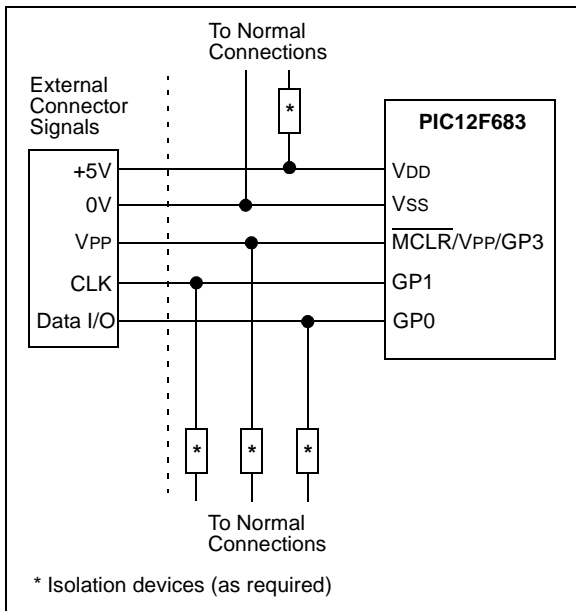
- clock
- data
- power
- ground
- programming voltage

This allows customers to manufacture boards with unprogrammed devices and then program the microcontroller just before shipping the product. This also allows the most recent firmware or a custom firmware to be programmed.

The device is placed into a Program/Verify mode by holding the GP0 and GP1 pins low, while raising the MCLR (VPP) pin from VIL to VIH. See the “PIC12F6XX/16F6XX Memory Programming Specification” (DS41204) for more information. GP0 becomes the programming data and GP1 becomes the programming clock. Both GP0 and GP1 are Schmitt Trigger inputs in Program/Verify mode.

A typical In-Circuit Serial Programming connection is shown in Figure 12-11.

**FIGURE 12-11: TYPICAL IN-CIRCUIT SERIAL PROGRAMMING CONNECTION**



## 12.11 In-Circuit Debugger

Since in-circuit debugging requires access to three pins, MPLAB® ICD 2 development with a 14-pin device is not practical. A special 14-pin PIC12F683 ICD device is used with MPLAB ICD 2 to provide separate clock, data and MCLR pins and frees all normally available pins to the user.

A special debugging adapter allows the ICD device to be used in place of a PIC12F683 device. The debugging adapter is the only source of the ICD device.

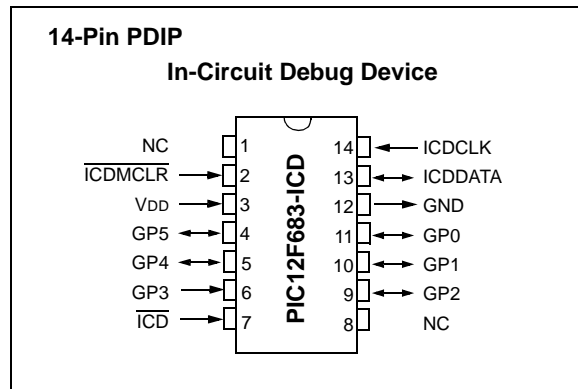
When the  $\overline{\text{ICD}}$  pin on the PIC12F683 ICD device is held low, the In-Circuit Debugger functionality is enabled. This function allows simple debugging functions when used with MPLAB ICD 2. When the microcontroller has this feature enabled, some of the resources are not available for general use. Table 12-9 shows which features are consumed by the background debugger.

**TABLE 12-9: DEBUGGER RESOURCES**

Resource	Description
Stack	1 level
Program Memory	Address 0h must be NOP 700h-7FFh

For more information, see “MPLAB® ICD 2 In-Circuit Debugger User's Guide” (DS15331), available on Microchip's web site ([www.microchip.com](http://www.microchip.com)).

**FIGURE 12-12: 14-PIN ICD PINOUT**





## 13.0 INSTRUCTION SET SUMMARY

The PIC12F683 instruction set is highly orthogonal and is comprised of three basic categories:

- **Byte-oriented** operations
- **Bit-oriented** operations
- **Literal and control** operations

Each PIC16 instruction is a 14-bit word divided into an **opcode**, which specifies the instruction type and one or more **operands**, which further specify the operation of the instruction. The formats for each of the categories is presented in Figure 13-1, while the various opcode fields are summarized in Table 13-1.

Table 13-2 lists the instructions recognized by the MPASM™ assembler.

For **byte-oriented** instructions, 'f' represents a file register designator and 'd' represents a destination designator. The file register designator specifies which file register is to be used by the instruction.

The destination designator specifies where the result of the operation is to be placed. If 'd' is zero, the result is placed in the W register. If 'd' is one, the result is placed in the file register specified in the instruction.

For **bit-oriented** instructions, 'b' represents a bit field designator, which selects the bit affected by the operation, while 'f' represents the address of the file in which the bit is located.

For **literal and control** operations, 'k' represents an 8-bit or 11-bit constant, or literal value.

One instruction cycle consists of four oscillator periods; for an oscillator frequency of 4 MHz, this gives a nominal instruction execution time of 1 μs. All instructions are executed within a single instruction cycle, unless a conditional test is true, or the program counter is changed as a result of an instruction. When this occurs, the execution takes two instruction cycles, with the second cycle executed as a NOP.

All instruction examples use the format '0xhh' to represent a hexadecimal number, where 'h' signifies a hexadecimal digit.

### 13.1 Read-Modify-Write Operations

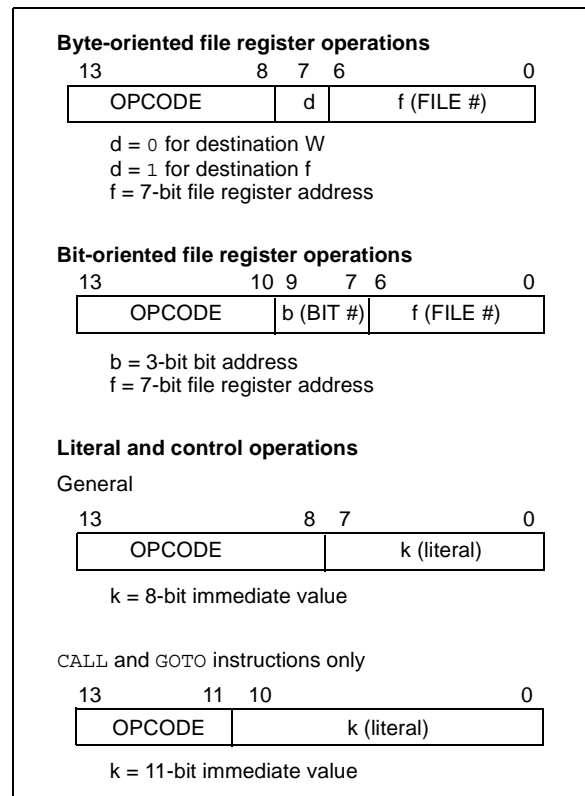
Any instruction that specifies a file register as part of the instruction performs a Read-Modify-Write (R-M-W) operation. The register is read, the data is modified, and the result is stored according to either the instruction, or the destination designator 'd'. A read operation is performed on a register even if the instruction writes to that register.

For example, a `CLRF PORTA` instruction will read PORTA, clear all the data bits, then write the result back to PORTA. This example would have the unintended consequence of clearing the condition that set the RAIF flag.

**TABLE 13-1: OPCODE FIELD DESCRIPTIONS**

Field	Description
f	Register file address (0x00 to 0x7F)
W	Working register (accumulator)
b	Bit address within an 8-bit file register
k	Literal field, constant data or label
x	Don't care location (= 0 or 1). The assembler will generate code with x = 0. It is the recommended form of use for compatibility with all Microchip software tools.
d	Destination select; d = 0: store result in W, d = 1: store result in file register f. Default is d = 1.
PC	Program Counter
TO	Time-out bit
C	Carry bit
DC	Digit carry bit
Z	Zero bit
PD	Power-down bit

**FIGURE 13-1: GENERAL FORMAT FOR INSTRUCTIONS**



# PIC12F683

**TABLE 13-2: PIC12F683 INSTRUCTION SET**

Mnemonic, Operands		Description	Cycles	14-Bit Opcode				Status Affected	Notes
				MSb		LSb			
BYTE-ORIENTED FILE REGISTER OPERATIONS									
ADDWF	f, d	Add W and f	1	00	0111	dfff	ffff	C, DC, Z	1, 2
ANDWF	f, d	AND W with f	1	00	0101	dfff	ffff	Z	1, 2
CLRF	f	Clear f	1	00	0001	1fff	ffff	Z	2
CLRW	—	Clear W	1	00	0001	0xxx	xxxx	Z	
COMF	f, d	Complement f	1	00	1001	dfff	ffff	Z	1, 2
DECf	f, d	Decrement f	1	00	0011	dfff	ffff	Z	1, 2
DECFSZ	f, d	Decrement f, Skip if 0	1(2)	00	1011	dfff	ffff		1, 2, 3
INCF	f, d	Increment f	1	00	1010	dfff	ffff	Z	1, 2
INCFSZ	f, d	Increment f, Skip if 0	1(2)	00	1111	dfff	ffff		1, 2, 3
IORWF	f, d	Inclusive OR W with f	1	00	0100	dfff	ffff	Z	1, 2
MOVF	f, d	Move f	1	00	1000	dfff	ffff	Z	1, 2
MOVWF	f	Move W to f	1	00	0000	1fff	ffff		
NOP	—	No Operation	1	00	0000	0xx0	0000		
RLF	f, d	Rotate Left f through Carry	1	00	1101	dfff	ffff	C	1, 2
RRF	f, d	Rotate Right f through Carry	1	00	1100	dfff	ffff	C	1, 2
SUBWF	f, d	Subtract W from f	1	00	0010	dfff	ffff	C, DC, Z	1, 2
SWAPF	f, d	Swap nibbles in f	1	00	1110	dfff	ffff		1, 2
XORWF	f, d	Exclusive OR W with f	1	00	0110	dfff	ffff	Z	1, 2
BIT-ORIENTED FILE REGISTER OPERATIONS									
BCF	f, b	Bit Clear f	1	01	00bb	bfff	ffff		1, 2
BSF	f, b	Bit Set f	1	01	01bb	bfff	ffff		1, 2
BTFSC	f, b	Bit Test f, Skip if Clear	1 (2)	01	10bb	bfff	ffff		3
BTFSS	f, b	Bit Test f, Skip if Set	1 (2)	01	11bb	bfff	ffff		3
LITERAL AND CONTROL OPERATIONS									
ADDLW	k	Add literal and W	1	11	111x	kkkk	kkkk	C, DC, Z	
ANDLW	k	AND literal with W	1	11	1001	kkkk	kkkk	Z	
CALL	k	Call Subroutine	2	10	0kkk	kkkk	kkkk		
CLRWDt	—	Clear Watchdog Timer	1	00	0000	0110	0100	$\overline{TO}$ , $\overline{PD}$	
GOTO	k	Go to address	2	10	1kkk	kkkk	kkkk		
IORLW	k	Inclusive OR literal with W	1	11	1000	kkkk	kkkk	Z	
MOVLW	k	Move literal to W	1	11	00xx	kkkk	kkkk		
RETFIE	—	Return from interrupt	2	00	0000	0000	1001		
RETLW	k	Return with literal in W	2	11	01xx	kkkk	kkkk		
RETURN	—	Return from Subroutine	2	00	0000	0000	1000		
SLEEP	—	Go into Standby mode	1	00	0000	0110	0011	$\overline{TO}$ , $\overline{PD}$	
SUBLW	k	Subtract W from literal	1	11	110x	kkkk	kkkk	C, DC, Z	
XORLW	k	Exclusive OR literal with W	1	11	1010	kkkk	kkkk	Z	

- Note 1:** When an I/O register is modified as a function of itself (e.g., `MOVF GPIO, 1`), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '0'.
- 2:** If this instruction is executed on the TMR0 register (and where applicable, d = 1), the prescaler will be cleared if assigned to the Timer0 module.
- 3:** If the Program Counter (PC) is modified, or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a NOP.

## 13.2 Instruction Descriptions

### ADDLW Add literal and W

Syntax:	[ <i>label</i> ] ADDLW <i>k</i>
Operands:	$0 \leq k \leq 255$
Operation:	$(W) + k \rightarrow (W)$
Status Affected:	C, DC, Z
Description:	The contents of the W register are added to the eight-bit literal 'k' and the result is placed in the W register.

### BCF Bit Clear f

Syntax:	[ <i>label</i> ] BCF <i>f</i> , <i>b</i>
Operands:	$0 \leq f \leq 127$ $0 \leq b \leq 7$
Operation:	$0 \rightarrow (f<b>)$
Status Affected:	None
Description:	Bit 'b' in register 'f' is cleared.

### ADDWF Add W and f

Syntax:	[ <i>label</i> ] ADDWF <i>f</i> , <i>d</i>
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	$(W) + (f) \rightarrow (\text{destination})$
Status Affected:	C, DC, Z
Description:	Add the contents of the W register with register 'f'. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f'.

### BSF Bit Set f

Syntax:	[ <i>label</i> ] BSF <i>f</i> , <i>b</i>
Operands:	$0 \leq f \leq 127$ $0 \leq b \leq 7$
Operation:	$1 \rightarrow (f<b>)$
Status Affected:	None
Description:	Bit 'b' in register 'f' is set.

### ANDLW AND literal with W

Syntax:	[ <i>label</i> ] ANDLW <i>k</i>
Operands:	$0 \leq k \leq 255$
Operation:	$(W) .\text{AND.} (k) \rightarrow (W)$
Status Affected:	Z
Description:	The contents of W register are AND'ed with the eight-bit literal 'k'. The result is placed in the W register.

### BTFSC Bit Test f, Skip if Clear

Syntax:	[ <i>label</i> ] BTFSC <i>f</i> , <i>b</i>
Operands:	$0 \leq f \leq 127$ $0 \leq b \leq 7$
Operation:	skip if $(f<b>) = 0$
Status Affected:	None
Description:	If bit 'b' in register 'f' is '1', the next instruction is executed. If bit 'b', in register 'f', is '0', the next instruction is discarded, and a NOP is executed instead, making this a 2-cycle instruction.

### ANDWF AND W with f

Syntax:	[ <i>label</i> ] ANDWF <i>f</i> , <i>d</i>
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	$(W) .\text{AND.} (f) \rightarrow (\text{destination})$
Status Affected:	Z
Description:	AND the W register with register 'f'. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f'.

# PIC12F683

---

**BTFSS**      **Bit Test f, Skip if Set**

---

Syntax:      [ *label* ] BTFSS f,b

Operands:     $0 \leq f \leq 127$   
               $0 \leq b < 7$

Operation:    skip if (f<b>) = 1

Status Affected: None

Description:   If bit 'b' in register 'f' is '0', the next instruction is executed.  
                  If bit 'b' is '1', then the next instruction is discarded and a NOP is executed instead, making this a 2-cycle instruction.

---

**CLRWDT**      **Clear Watchdog Timer**

---

Syntax:      [ *label* ] CLRWDT

Operands:    None

Operation:    00h → WDT  
              0 → WDT prescaler,  
              1 →  $\overline{TO}$   
              1 →  $\overline{PD}$

Status Affected:  $\overline{TO}$ ,  $\overline{PD}$

Description:   CLRWDT instruction resets the Watchdog Timer. It also resets the prescaler of the WDT.  
                  Status bits  $\overline{TO}$  and  $\overline{PD}$  are set.

---

**CALL**      **Call Subroutine**

---

Syntax:      [ *label* ] CALL k

Operands:     $0 \leq k \leq 2047$

Operation:    (PC)+1 → TOS,  
              k → PC<10:0>,  
              (PCLATH<4:3>) → PC<12:11>

Status Affected: None

Description:   Call Subroutine. First, return address (PC + 1) is pushed onto the stack. The eleven-bit immediate address is loaded into PC bits <10:0>. The upper bits of the PC are loaded from PCLATH. CALL is a two-cycle instruction.

---

**COMF**      **Complement f**

---

Syntax:      [ *label* ] COMF f,d

Operands:     $0 \leq f \leq 127$   
              d ∈ [0,1]

Operation:    ( $\bar{f}$ ) → (destination)

Status Affected: Z

Description:   The contents of register 'f' are complemented. If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in register 'f'.

---

**CLRF**      **Clear f**

---

Syntax:      [ *label* ] CLRF f

Operands:     $0 \leq f \leq 127$

Operation:    00h → (f)  
              1 → Z

Status Affected: Z

Description:   The contents of register 'f' are cleared and the Z bit is set.

---

**DECF**      **Decrement f**

---

Syntax:      [ *label* ] DECF f,d

Operands:     $0 \leq f \leq 127$   
              d ∈ [0,1]

Operation:    (f) - 1 → (destination)

Status Affected: Z

Description:   Decrement register 'f'. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f'.

---

**CLRW**      **Clear W**

---

Syntax:      [ *label* ] CLRW

Operands:    None

Operation:    00h → (W)  
              1 → Z

Status Affected: Z

Description:   W register is cleared. Zero bit (Z) is set.

## DECFSZ      Decrement f, Skip if 0

**Syntax:**      [ *label* ] DECFSZ f,d

**Operands:**       $0 \leq f \leq 127$   
 $d \in [0,1]$

**Operation:**       $(f) - 1 \rightarrow (\text{destination});$   
skip if result = 0

**Status Affected:**      None

**Description:**      The contents of register 'f' are decremented. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'.  
If the result is '1', the next instruction is executed. If the result is '0', then a NOP is executed instead, making it a 2-cycle instruction.

## INCFSZ      Increment f, Skip if 0

**Syntax:**      [ *label* ] INCFSZ f,d

**Operands:**       $0 \leq f \leq 127$   
 $d \in [0,1]$

**Operation:**       $(f) + 1 \rightarrow (\text{destination});$   
skip if result = 0

**Status Affected:**      None

**Description:**      The contents of register 'f' are incremented. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'.  
If the result is '1', the next instruction is executed. If the result is '0', a NOP is executed instead, making it a 2-cycle instruction.

## GOTO      Unconditional Branch

**Syntax:**      [ *label* ] GOTO k

**Operands:**       $0 \leq k \leq 2047$

**Operation:**       $k \rightarrow \text{PC}<10:0>$   
 $\text{PCLATH}<4:3> \rightarrow \text{PC}<12:11>$

**Status Affected:**      None

**Description:**      GOTO is an unconditional branch. The eleven-bit immediate value is loaded into PC bits <10:0>. The upper bits of PC are loaded from PCLATH<4:3>. GOTO is a two-cycle instruction.

## IORLW      Inclusive OR literal with W

**Syntax:**      [ *label* ] IORLW k

**Operands:**       $0 \leq k \leq 255$

**Operation:**       $(W) .\text{OR. } k \rightarrow (W)$

**Status Affected:**      Z

**Description:**      The contents of the W register are OR'ed with the eight-bit literal 'k'. The result is placed in the W register.

## INCF      Increment f

**Syntax:**      [ *label* ] INCF f,d

**Operands:**       $0 \leq f \leq 127$   
 $d \in [0,1]$

**Operation:**       $(f) + 1 \rightarrow (\text{destination})$

**Status Affected:**      Z

**Description:**      The contents of register 'f' are incremented. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'.

## IORWF      Inclusive OR W with f

**Syntax:**      [ *label* ] IORWF f,d

**Operands:**       $0 \leq f \leq 127$   
 $d \in [0,1]$

**Operation:**       $(W) .\text{OR. } (f) \rightarrow (\text{destination})$

**Status Affected:**      Z

**Description:**      Inclusive OR the W register with register 'f'. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'.

# PIC12F683

<b>MOVF</b>	<b>Move f</b>
Syntax:	[ <i>label</i> ] MOVF f,d
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	(f) $\rightarrow$ (dest)
Status Affected:	Z
Description:	The contents of register f is moved to a destination dependent upon the status of d. If d = 0, destination is W register. If d = 1, the destination is file register f itself. d = 1 is useful to test a file register since status flag Z is affected.
Words:	1
Cycles:	1
Example:	<pre>MOVF    FSR, 0</pre> <p>After Instruction</p> <p>W = value in FSR register</p> <p>Z = 1</p>

<b>MOVLW</b>	<b>Move literal to W</b>
Syntax:	[ <i>label</i> ] MOVLW k
Operands:	$0 \leq k \leq 255$
Operation:	k $\rightarrow$ (W)
Status Affected:	None
Description:	The eight-bit literal 'k' is loaded into W register. The "don't cares" will assemble as '0's.
Words:	1
Cycles:	1
Example:	<pre>MOVLW    0x5A</pre> <p>After Instruction</p> <p>W = 0x5A</p>

<b>MOVWF</b>	<b>Move W to f</b>
Syntax:	[ <i>label</i> ] MOVWF f
Operands:	$0 \leq f \leq 127$
Operation:	(W) $\rightarrow$ (f)
Status Affected:	None
Description:	Move data from W register to register 'f'.
Words:	1
Cycles:	1
Example:	<pre>MOVW    OPTION F</pre> <p>Before Instruction</p> <p>OPTION = 0xFF W = 0x4F</p> <p>After Instruction</p> <p>OPTION = 0x4F W = 0x4F</p>

<b>NOP</b>	<b>No Operation</b>
Syntax:	[ <i>label</i> ] NOP
Operands:	None
Operation:	No operation
Status Affected:	None
Description:	No operation.
Words:	1
Cycles:	1
Example:	<pre>NOP</pre>

<b>RETFIE</b>	<b>Return from Interrupt</b>
Syntax:	[ <i>label</i> ] RETFIE
Operands:	None
Operation:	TOS → PC, 1 → GIE
Status Affected:	None
Description:	Return from Interrupt. Stack is POPed and Top-of-Stack (TOS) is loaded in the PC. Interrupts are enabled by setting Global Interrupt Enable bit, GIE (INTCON<7>). This is a two-cycle instruction.
Words:	1
Cycles:	2
<u>Example:</u>	RETFIE After Interrupt PC = TOS GIE = 1

<b>RETLW</b>	<b>Return with literal in W</b>
Syntax:	[ <i>label</i> ] RETLW k
Operands:	0 ≤ k ≤ 255
Operation:	k → (W); TOS → PC
Status Affected:	None
Description:	The W register is loaded with the eight bit literal 'k'. The program counter is loaded from the top of the stack (the return address). This is a two-cycle instruction.
Words:	1
Cycles:	2
<u>Example:</u>	CALL TABLE;W contains table ;offset value • ;W now has table value • • ADDWF PC ;W = offset RETLW k1 ;Begin table RETLW k2 ; • • • RETLW kn ; End of table  Before Instruction W = 0x07 After Instruction W = value of k8

<b>RETURN</b>	<b>Return from Subroutine</b>
Syntax:	[ <i>label</i> ] RETURN
Operands:	None
Operation:	TOS → PC
Status Affected:	None
Description:	Return from subroutine. The stack is POPed and the top of the stack (TOS) is loaded into the program counter. This is a two-cycle instruction.

# PIC12F683

## RLF Rotate Left f through Carry

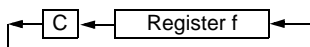
**Syntax:** [ *label* ] RLF f,d

**Operands:**  $0 \leq f \leq 127$   
 $d \in [0,1]$

**Operation:** See description below

**Status Affected:** C

**Description:** The contents of register 'f' are rotated one bit to the left through the Carry flag. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is stored back in register 'f'.



**Words:** 1

**Cycles:** 1

**Example:** RLF REG1,0

Before Instruction

REG1 = 1110 0110  
 C = 0

After Instruction

REG1 = 1110 0110  
 W = 1100 1100  
 C = 1

## SLEEP Enter Sleep mode

**Syntax:** [ *label* ] SLEEP

**Operands:** None

**Operation:** 00h → WDT,  
 0 → WDT prescaler,  
 1 →  $\overline{TO}$ ,  
 0 →  $\overline{PD}$

**Status Affected:**  $\overline{TO}$ ,  $\overline{PD}$

**Description:** The power-down Status bit,  $\overline{PD}$  is cleared. Time-out Status bit,  $\overline{TO}$  is set. Watchdog Timer and its prescaler are cleared. The processor is put into Sleep mode with the oscillator stopped.

## RRF Rotate Right f through Carry

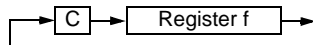
**Syntax:** [ *label* ] RRF f,d

**Operands:**  $0 \leq f \leq 127$   
 $d \in [0,1]$

**Operation:** See description below

**Status Affected:** C

**Description:** The contents of register 'f' are rotated one bit to the right through the Carry flag. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'.



## SUBLW Subtract W from literal

**Syntax:** [ *label* ] SUBLW k

**Operands:**  $0 \leq k \leq 255$

**Operation:**  $k - (W) \rightarrow (W)$

**Status Affected:** C, DC, Z

**Description:** The W register is subtracted (2's complement method) from the eight-bit literal 'k'. The result is placed in the W register.

C = 0	$W > k$
C = 1	$W \leq k$
DC = 0	$W<3:0> > k<3:0>$
DC = 1	$W<3:0> \leq k<3:0>$



## **SUBWF**      **Subtract W from f**

Syntax:      [ *label* ] SUBWF f,d

Operands:     $0 \leq f \leq 127$   
                   $d \in [0,1]$

Operation:     $(f) - (W) \rightarrow (\text{destination})$

Status Affected: C, DC, Z

Description:   Subtract (2's complement method) W register from register 'f'. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f'.

C = 0	$W > f$
C = 1	$W \leq f$
DC = 0	$W<3:0> > f<3:0>$
DC = 1	$W<3:0> \leq f<3:0>$

## **XORLW**      **Exclusive OR literal with W**

Syntax:      [ *label* ] XORLW k

Operands:     $0 \leq k \leq 255$

Operation:     $(W) .XOR. k \rightarrow (W)$

Status Affected: Z

Description:   The contents of the W register are XOR'ed with the eight-bit literal 'k'. The result is placed in the W register.

## **SWAPF**      **Swap Nibbles in f**

Syntax:      [ *label* ] SWAPF f,d

Operands:     $0 \leq f \leq 127$   
                   $d \in [0,1]$

Operation:     $(f<3:0>) \rightarrow (\text{destination}<7:4>),$   
                   $(f<7:4>) \rightarrow (\text{destination}<3:0>)$

Status Affected: None

Description:   The upper and lower nibbles of register 'f' are exchanged. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed in register 'f'.

## **XORWF**      **Exclusive OR W with f**

Syntax:      [ *label* ] XORWF f,d

Operands:     $0 \leq f \leq 127$   
                   $d \in [0,1]$

Operation:     $(W) .XOR. (f) \rightarrow (\text{destination})$

Status Affected: Z

Description:   Exclusive OR the contents of the W register with register 'f'. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f'.

# PIC12F683

---

NOTES:

## 14.0 DEVELOPMENT SUPPORT

The PIC® microcontrollers are supported with a full range of hardware and software development tools:

- Integrated Development Environment
  - MPLAB® IDE Software
- Assemblers/Compilers/Linkers
  - MPASM™ Assembler
  - MPLAB C18 and MPLAB C30 C Compilers
  - MPLINK™ Object Linker/  
MPLIB™ Object Librarian
  - MPLAB ASM30 Assembler/Linker/Library
- Simulators
  - MPLAB SIM Software Simulator
- Emulators
  - MPLAB ICE 2000 In-Circuit Emulator
  - MPLAB REAL ICE™ In-Circuit Emulator
- In-Circuit Debugger
  - MPLAB ICD 2
- Device Programmers
  - PICSTART® Plus Development Programmer
  - MPLAB PM3 Device Programmer
  - PICkit™ 2 Development Programmer
- Low-Cost Demonstration and Development Boards and Evaluation Kits

## 14.1 MPLAB Integrated Development Environment Software

The MPLAB IDE software brings an ease of software development previously unseen in the 8/16-bit microcontroller market. The MPLAB IDE is a Windows® operating system-based application that contains:

- A single graphical interface to all debugging tools
  - Simulator
  - Programmer (sold separately)
  - Emulator (sold separately)
  - In-Circuit Debugger (sold separately)
- A full-featured editor with color-coded context
- A multiple project manager
- Customizable data windows with direct edit of contents
- High-level source code debugging
- Visual device initializer for easy register initialization
- Mouse over variable inspection
- Drag and drop variables from source to watch windows
- Extensive on-line help
- Integration of select third party tools, such as HI-TECH Software C Compilers and IAR C Compilers

The MPLAB IDE allows you to:

- Edit your source files (either assembly or C)
- One touch assemble (or compile) and download to PIC MCU emulator and simulator tools (automatically updates all project information)
- Debug using:
  - Source files (assembly or C)
  - Mixed assembly and C
  - Machine code

MPLAB IDE supports multiple debugging tools in a single development paradigm, from the cost-effective simulators, through low-cost in-circuit debuggers, to full-featured emulators. This eliminates the learning curve when upgrading to tools with increased flexibility and power.

## 14.2 MPASM Assembler

The MPASM Assembler is a full-featured, universal macro assembler for all PIC MCUs.

The MPASM Assembler generates relocatable object files for the MPLINK Object Linker, Intel® standard HEX files, MAP files to detail memory usage and symbol reference, absolute LST files that contain source lines and generated machine code and COFF files for debugging.

The MPASM Assembler features include:

- Integration into MPLAB IDE projects
- User-defined macros to streamline assembly code
- Conditional assembly for multi-purpose source files
- Directives that allow complete control over the assembly process

## 14.3 MPLAB C18 and MPLAB C30 C Compilers

The MPLAB C18 and MPLAB C30 Code Development Systems are complete ANSI C compilers for Microchip's PIC18 and PIC24 families of microcontrollers and the dsPIC30 and dsPIC33 family of digital signal controllers. These compilers provide powerful integration capabilities, superior code optimization and ease of use not found with other compilers.

For easy source level debugging, the compilers provide symbol information that is optimized to the MPLAB IDE debugger.

## 14.4 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK Object Linker combines relocatable objects created by the MPASM Assembler and the MPLAB C18 C Compiler. It can link relocatable objects from precompiled libraries, using directives from a linker script.

The MPLIB Object Librarian manages the creation and modification of library files of precompiled code. When a routine from a library is called from a source file, only the modules that contain that routine will be linked in with the application. This allows large libraries to be used efficiently in many different applications.

The object linker/library features include:

- Efficient linking of single libraries instead of many smaller files
- Enhanced code maintainability by grouping related modules together
- Flexible creation of libraries with easy module listing, replacement, deletion and extraction

## 14.5 MPLAB ASM30 Assembler, Linker and Librarian

MPLAB ASM30 Assembler produces relocatable machine code from symbolic assembly language for dsPIC30F devices. MPLAB C30 C Compiler uses the assembler to produce its object file. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. Notable features of the assembler include:

- Support for the entire dsPIC30F instruction set
- Support for fixed-point and floating-point data
- Command line interface
- Rich directive set
- Flexible macro language
- MPLAB IDE compatibility

## 14.6 MPLAB SIM Software Simulator

The MPLAB SIM Software Simulator allows code development in a PC-hosted environment by simulating the PIC MCUs and dsPIC® DSCs on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a comprehensive stimulus controller. Registers can be logged to files for further run-time analysis. The trace buffer and logic analyzer display extend the power of the simulator to record and track program execution, actions on I/O, most peripherals and internal registers.

The MPLAB SIM Software Simulator fully supports symbolic debugging using the MPLAB C18 and MPLAB C30 C Compilers, and the MPASM and MPLAB ASM30 Assemblers. The software simulator offers the flexibility to develop and debug code outside of the hardware laboratory environment, making it an excellent, economical software development tool.

## 14.7 MPLAB ICE 2000 High-Performance In-Circuit Emulator

The MPLAB ICE 2000 In-Circuit Emulator is intended to provide the product development engineer with a complete microcontroller design tool set for PIC microcontrollers. Software control of the MPLAB ICE 2000 In-Circuit Emulator is advanced by the MPLAB Integrated Development Environment, which allows editing, building, downloading and source debugging from a single environment.

The MPLAB ICE 2000 is a full-featured emulator system with enhanced trace, trigger and data monitoring features. Interchangeable processor modules allow the system to be easily reconfigured for emulation of different processors. The architecture of the MPLAB ICE 2000 In-Circuit Emulator allows expansion to support new PIC microcontrollers.

The MPLAB ICE 2000 In-Circuit Emulator system has been designed as a real-time emulation system with advanced features that are typically found on more expensive development tools. The PC platform and Microsoft® Windows® 32-bit operating system were chosen to best make these features available in a simple, unified application.

## 14.8 MPLAB REAL ICE In-Circuit Emulator System

MPLAB REAL ICE In-Circuit Emulator System is Microchip's next generation high-speed emulator for Microchip Flash DSC® and MCU devices. It debugs and programs PIC® and dsPIC® Flash microcontrollers with the easy-to-use, powerful graphical user interface of the MPLAB Integrated Development Environment (IDE), included with each kit.

The MPLAB REAL ICE probe is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with either a connector compatible with the popular MPLAB ICD 2 system (RJ11) or with the new high speed, noise tolerant, low-voltage differential signal (LVDS) interconnection (CAT5).

MPLAB REAL ICE is field upgradeable through future firmware downloads in MPLAB IDE. In upcoming releases of MPLAB IDE, new devices will be supported, and new features will be added, such as software breakpoints and assembly code trace. MPLAB REAL ICE offers significant advantages over competitive emulators including low-cost, full-speed emulation, real-time variable watches, trace analysis, complex breakpoints, a ruggedized probe interface and long (up to three meters) interconnection cables.

## 14.9 MPLAB ICD 2 In-Circuit Debugger

Microchip's In-Circuit Debugger, MPLAB ICD 2, is a powerful, low-cost, run-time development tool, connecting to the host PC via an RS-232 or high-speed USB interface. This tool is based on the Flash PIC MCUs and can be used to develop for these and other PIC MCUs and dsPIC DSCs. The MPLAB ICD 2 utilizes the in-circuit debugging capability built into the Flash devices. This feature, along with Microchip's In-Circuit Serial Programming™ (ICSP™) protocol, offers cost-effective, in-circuit Flash debugging from the graphical user interface of the MPLAB Integrated Development Environment. This enables a designer to develop and debug source code by setting breakpoints, single stepping and watching variables, and CPU status and peripheral registers. Running at full speed enables testing hardware and applications in real time. MPLAB ICD 2 also serves as a development programmer for selected PIC devices.

## 14.10 MPLAB PM3 Device Programmer

The MPLAB PM3 Device Programmer is a universal, CE compliant device programmer with programmable voltage verification at VDDMIN and VDDMAX for maximum reliability. It features a large LCD display (128 x 64) for menus and error messages and a modular, detachable socket assembly to support various package types. The ICSP™ cable assembly is included as a standard item. In Stand-Alone mode, the MPLAB PM3 Device Programmer can read, verify and program PIC devices without a PC connection. It can also set code protection in this mode. The MPLAB PM3 connects to the host PC via an RS-232 or USB cable. The MPLAB PM3 has high-speed communications and optimized algorithms for quick programming of large memory devices and incorporates an SD/MMC card for file storage and secure data applications.

## 14.11 PICSTART Plus Development Programmer

The PICSTART Plus Development Programmer is an easy-to-use, low-cost, prototype programmer. It connects to the PC via a COM (RS-232) port. MPLAB Integrated Development Environment software makes using the programmer simple and efficient. The PICSTART Plus Development Programmer supports most PIC devices in DIP packages up to 40 pins. Larger pin count devices, such as the PIC16C92X and PIC17C76X, may be supported with an adapter socket. The PICSTART Plus Development Programmer is CE compliant.

## 14.12 PICkit 2 Development Programmer

The PICkit™ 2 Development Programmer is a low-cost programmer and selected Flash device debugger with an easy-to-use interface for programming many of Microchip's baseline, mid-range and PIC18F families of Flash memory microcontrollers. The PICkit 2 Starter Kit includes a prototyping development board, twelve sequential lessons, software and HI-TECH's PICC™ Lite C compiler, and is designed to help get up to speed quickly using PIC® microcontrollers. The kit provides everything needed to program, evaluate and develop applications using Microchip's powerful, mid-range Flash memory family of microcontrollers.

## 14.13 Demonstration, Development and Evaluation Boards

A wide variety of demonstration, development and evaluation boards for various PIC MCUs and dsPIC DSCs allows quick application development on fully functional systems. Most boards include prototyping areas for adding custom circuitry and provide application firmware and source code for examination and modification.

The boards support a variety of features, including LEDs, temperature sensors, switches, speakers, RS-232 interfaces, LCD displays, potentiometers and additional EEPROM memory.

The demonstration and development boards can be used in teaching environments, for prototyping custom circuits and for learning about various microcontroller applications.

In addition to the PICDEM™ and dsPICDEM™ demonstration/development board series of circuits, Microchip has a line of evaluation kits and demonstration software for analog filter design, KEELOQ® security ICs, CAN, IrDA®, PowerSmart® battery management, SEEVAL® evaluation system, Sigma-Delta ADC, flow rate sensing, plus many more.

Check the Microchip web page ([www.microchip.com](http://www.microchip.com)) and the latest "Product Selector Guide" (DS00148) for the complete list of demonstration, development and evaluation kits.

## 15.0 ELECTRICAL SPECIFICATIONS

### Absolute Maximum Ratings<sup>(†)</sup>

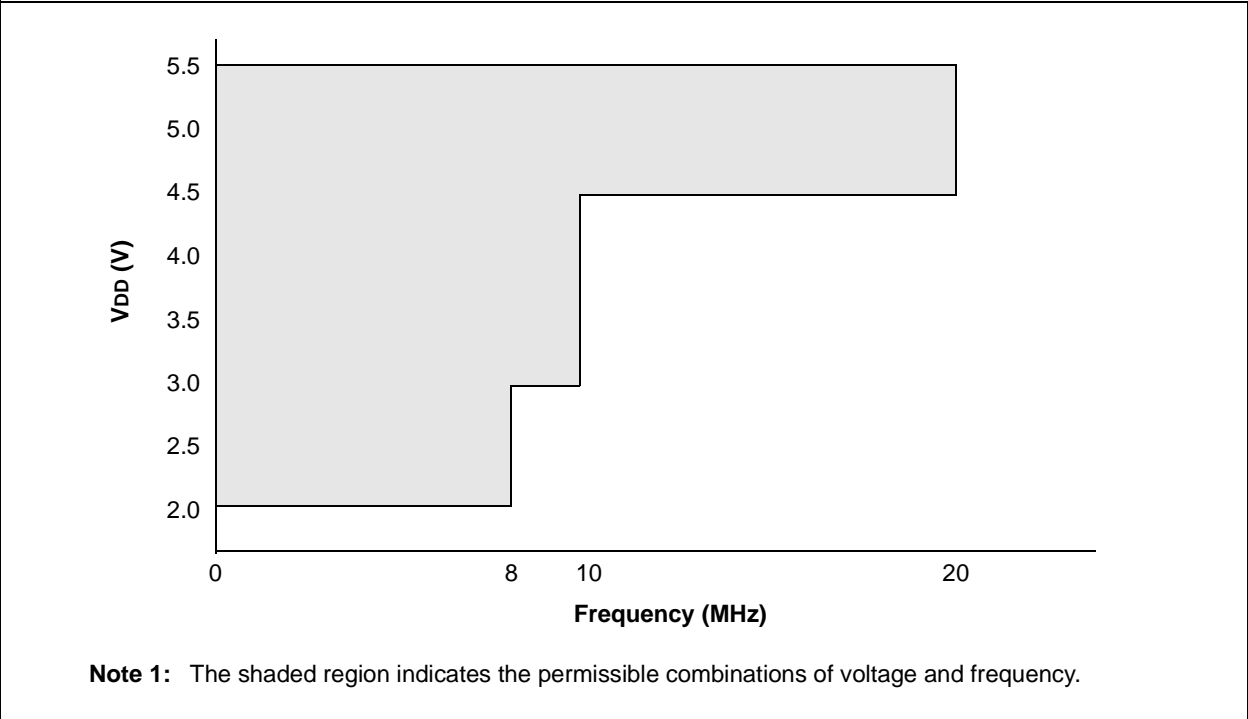
Ambient temperature under bias .....	-40° to +125°C
Storage temperature .....	-65°C to +150°C
Voltage on VDD with respect to VSS .....	-0.3V to +6.5V
Voltage on $\overline{\text{MCLR}}$ with respect to VSS .....	-0.3V to +13.5V
Voltage on all other pins with respect to VSS .....	-0.3V to (VDD + 0.3V)
Total power dissipation <sup>(1)</sup> .....	800 mW
Maximum current out of VSS pin .....	95 mA
Maximum current into VDD pin .....	95 mA
Input clamp current, I <sub>IK</sub> (V <sub>I</sub> < 0 or V <sub>I</sub> > VDD) .....	± 20 mA
Output clamp current, I <sub>OK</sub> (V <sub>O</sub> < 0 or V <sub>O</sub> > VDD) .....	± 20 mA
Maximum output current sunk by any I/O pin .....	25 mA
Maximum output current sourced by any I/O pin .....	25 mA
Maximum current sunk by GPIO .....	90 mA
Maximum current sourced GPIO .....	90 mA

**Note 1:** Power dissipation is calculated as follows:  $P_{DIS} = V_{DD} \times \{I_{DD} - \sum I_{OH}\} + \sum \{(V_{DD} - V_{OH}) \times I_{OH}\} + \sum (V_{OL} \times I_{OL})$ .

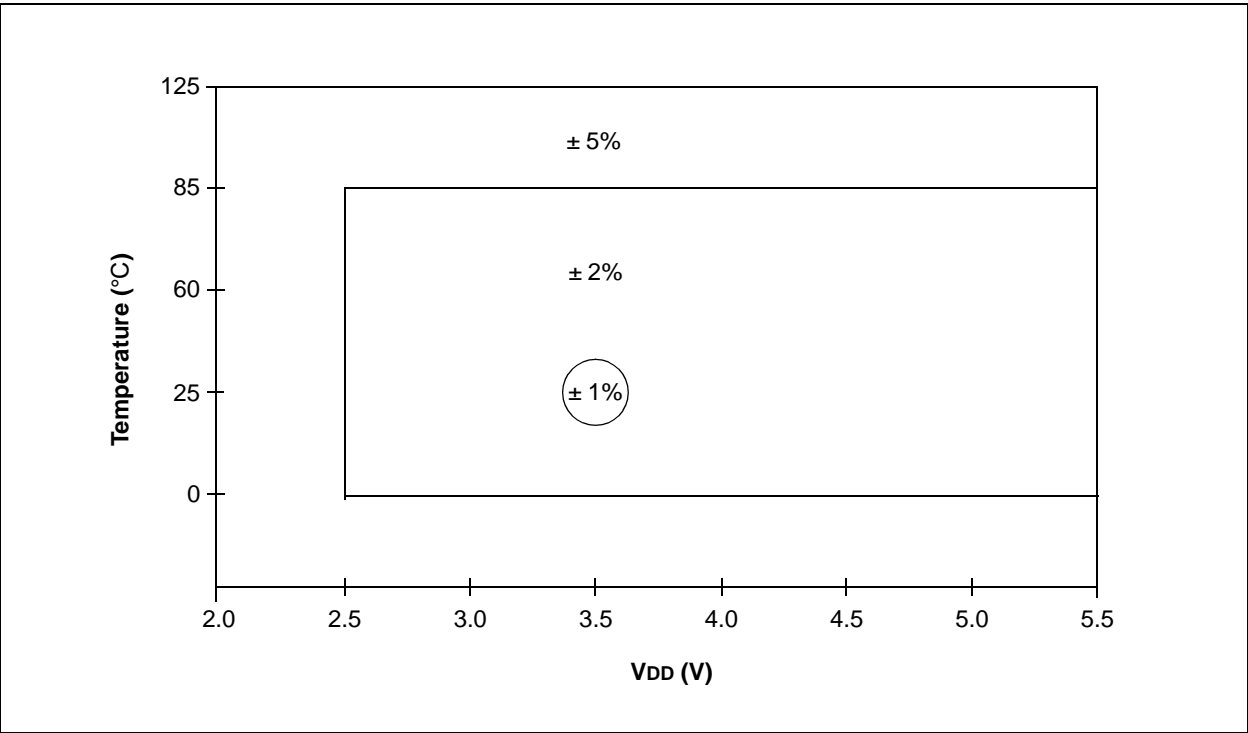
† NOTICE: Stresses above those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure above maximum rating conditions for extended periods may affect device reliability.

# PIC12F683

**FIGURE 15-1: PIC12F683 VOLTAGE-FREQUENCY GRAPH,  $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$**



**FIGURE 15-2: HFINTOSC FREQUENCY ACCURACY OVER DEVICE V<sub>DD</sub> AND TEMPERATURE**





## 15.1 DC Characteristics: PIC12F683-I (Industrial) PIC12F683-E (Extended)

DC CHARACTERISTICS			Standard Operating Conditions (unless otherwise stated)				
			Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended				
Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
D001 D001C D001D	VDD	Supply Voltage	2.0 2.0 3.0 4.5	— — — —	5.5 5.5 5.5 5.5	V V V V	FOSC ≤ 8 MHz: HFINTOSC, EC FOSC ≤ 4 MHz FOSC ≤ 10 MHz FOSC ≤ 20 MHz
D002*	VDR	RAM Data Retention Voltage <sup>(1)</sup>	1.5	—	—	V	Device in Sleep mode
D003	VPOR	VDD Start Voltage to ensure internal Power-on Reset signal	—	VSS	—	V	See Section 12.3.1 “Power-on Reset” for details.
D004*	SVDD	VDD Rise Rate to ensure internal Power-on Reset signal	0.05	—	—	V/ms	See Section 12.3.1 “Power-on Reset” for details.

\* These parameters are characterized but not tested.

† Data in “Typ” column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** This is the limit to which VDD can be lowered in Sleep mode without losing RAM data.

# PIC12F683

## 15.2 DC Characteristics: PIC12F683-I (Industrial) PIC12F683-E (Extended)

DC CHARACTERISTICS		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended					
Param No.	Device Characteristics	Min	Typ†	Max	Units	Conditions	
						VDD	Note
D010	Supply Current (IDD) <sup>(1, 2)</sup>	—	11	16	μA	2.0	Fosc = 32 kHz
		—	18	28	μA	3.0	LP Oscillator mode
		—	35	54	μA	5.0	
D011*		—	140	240	μA	2.0	Fosc = 1 MHz
		—	220	380	μA	3.0	XT Oscillator mode
		—	380	550	μA	5.0	
D012		—	260	360	μA	2.0	Fosc = 4 MHz
		—	420	650	μA	3.0	XT Oscillator mode
		—	0.8	1.1	mA	5.0	
D013*		—	130	220	μA	2.0	Fosc = 1 MHz
		—	215	360	μA	3.0	EC Oscillator mode
		—	360	520	μA	5.0	
D014		—	220	340	μA	2.0	Fosc = 4 MHz
		—	375	550	μA	3.0	EC Oscillator mode
		—	0.65	1.0	mA	5.0	
D015		—	8	20	μA	2.0	Fosc = 31 kHz
		—	16	40	μA	3.0	LFINTOSC mode
		—	31	65	μA	5.0	
D016*		—	340	450	μA	2.0	Fosc = 4 MHz
		—	500	700	μA	3.0	HFINTOSC mode
		—	0.8	1.2	mA	5.0	
D017		—	410	650	μA	2.0	Fosc = 8 MHz
		—	700	950	μA	3.0	HFINTOSC mode
		—	1.30	1.65	mA	5.0	
D018		—	230	400	μA	2.0	Fosc = 4 MHz
		—	400	680	μA	3.0	EXTRC mode <sup>(3)</sup>
		—	0.63	1.1	mA	5.0	
D019		—	2.6	3.25	mA	4.5	Fosc = 20 MHz
		—	2.8	3.35	mA	5.0	HS Oscillator mode

\* These parameters are characterized but not tested.

† Data in "Typ" column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

- Note 1:** The test conditions for all IDD measurements in active operation mode are: OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to VDD; MCLR = VDD; WDT disabled.
- 2:** The supply current is mainly a function of the operating voltage and frequency. Other factors, such as I/O pin loading and switching rate, oscillator type, internal code execution pattern and temperature, also have an impact on the current consumption.
- 3:** For RC oscillator configurations, current through REXT is not included. The current through the resistor can be extended by the formula  $I_R = V_{DD}/2R_{EXT}$  (mA) with REXT in kΩ.

## 15.3 DC Characteristics: PIC12F683-I (Industrial)

DC CHARACTERISTICS		Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial					
Param No.	Device Characteristics	Min	Typ†	Max	Units	Conditions	
						VDD	Note
D020	Power-down Base Current(I <sub>PD</sub> ) <sup>(2)</sup>	—	0.05	1.2	μA	2.0	WDT, BOR, Comparators, VREF and T1OSC disabled
		—	0.15	1.5	μA	3.0	
		—	0.35	1.8	μA	5.0	
		—	150	500	nA	3.0	$-40^{\circ}\text{C} \leq T_A \leq +25^{\circ}\text{C}$
D021		—	1.0	2.2	μA	2.0	WDT Current <sup>(1)</sup>
		—	2.0	4.0	μA	3.0	
		—	3.0	7.0	μA	5.0	
D022		—	42	60	μA	3.0	BOR Current <sup>(1)</sup>
		—	85	122	μA	5.0	
D023		—	32	45	μA	2.0	Comparator Current <sup>(1)</sup> , both comparators enabled
		—	60	78	μA	3.0	
		—	120	160	μA	5.0	
D024		—	30	36	μA	2.0	CVREF Current <sup>(1)</sup> (high range)
		—	45	55	μA	3.0	
		—	75	95	μA	5.0	
D025*		—	39	47	μA	2.0	CVREF Current <sup>(1)</sup> (low range)
		—	59	72	μA	3.0	
		—	98	124	μA	5.0	
D026		—	4.5	7.0	μA	2.0	T1OSC Current <sup>(1)</sup> , 32.768 kHz
		—	5.0	8.0	μA	3.0	
		—	6.0	12	μA	5.0	
D027		—	0.30	1.6	μA	3.0	A/D Current <sup>(1)</sup> , no conversion in progress
		—	0.36	1.9	μA	5.0	

\* These parameters are characterized but not tested.

† Data in "Typ" column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

- Note 1:** The peripheral current is the sum of the base I<sub>DD</sub> or I<sub>PD</sub> and the additional current consumed when this peripheral is enabled. The peripheral Δ current can be determined by subtracting the base I<sub>DD</sub> or I<sub>PD</sub> current from this limit. Max values should be used when calculating total current consumption.
- 2:** The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to VDD.

# PIC12F683

## 15.4 DC Characteristics: PIC12F683-E (Extended)

DC CHARACTERISTICS		Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ for extended					
Param No.	Device Characteristics	Min	Typ†	Max	Units	Conditions	
						V <sub>DD</sub>	Note
D020E	Power-down Base Current (I <sub>PD</sub> ) <sup>(2)</sup>	—	0.05	9	μA	2.0	WDT, BOR, Comparators, V <sub>REF</sub> and T1OSC disabled
		—	0.15	11	μA	3.0	
		—	0.35	15	μA	5.0	
D021E		—	1	17.5	μA	2.0	WDT Current <sup>(1)</sup>
		—	2	19	μA	3.0	
		—	3	22	μA	5.0	
D022E		—	42	65	μA	3.0	BOR Current <sup>(1)</sup>
		—	85	127	μA	5.0	
D023E		—	32	45	μA	2.0	Comparator Current <sup>(1)</sup> , both comparators enabled
		—	60	78	μA	3.0	
		—	120	160	μA	5.0	
D024E		—	30	70	μA	2.0	CV <sub>REF</sub> Current <sup>(1)</sup> (high range)
		—	45	90	μA	3.0	
		—	75	120	μA	5.0	
D025E*		—	39	91	μA	2.0	CV <sub>REF</sub> Current <sup>(1)</sup> (low range)
		—	59	117	μA	3.0	
		—	98	156	μA	5.0	
D026E		—	4.5	25	μA	2.0	T1OSC Current <sup>(1)</sup> , 32.768 kHz
		—	5	30	μA	3.0	
		—	6	40	μA	5.0	
D027E		—	0.30	12	μA	3.0	A/D Current <sup>(1)</sup> , no conversion in progress
		—	0.36	16	μA	5.0	

\* These parameters are characterized but not tested.

† Data in "Typ" column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

- Note 1:** The peripheral current is the sum of the base I<sub>DD</sub> or I<sub>PD</sub> and the additional current consumed when this peripheral is enabled. The peripheral Δ current can be determined by subtracting the base I<sub>DD</sub> or I<sub>PD</sub> current from this limit. Max values should be used when calculating total current consumption.
- 2:** The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to V<sub>DD</sub>.

## 15.5 DC Characteristics: PIC12F683-I (Industrial) PIC12F683-E (Extended)

DC CHARACTERISTICS			Standard Operating Conditions (unless otherwise stated)				
			Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ for extended				
Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
D030	V <sub>IL</sub>	<b>Input Low Voltage</b> I/O Port: with TTL buffer	V <sub>SS</sub>	—	0.8	V	$4.5\text{V} \leq V_{DD} \leq 5.5\text{V}$
D030A			V <sub>SS</sub>	—	$0.15 V_{DD}$	V	$2.0\text{V} \leq V_{DD} \leq 4.5\text{V}$
D031		with Schmitt Trigger buffer	V <sub>SS</sub>	—	$0.2 V_{DD}$	V	$2.0\text{V} \leq V_{DD} \leq 5.5\text{V}$
D032		MCLR, OSC1 (RC mode) <sup>(1)</sup>	V <sub>SS</sub>	—	$0.2 V_{DD}$	V	
D033		OSC1 (XT and LP modes)	V <sub>SS</sub>	—	0.3	V	
D033A		OSC1 (HS mode)	V <sub>SS</sub>	—	$0.3 V_{DD}$	V	
D040	V <sub>IH</sub>	<b>Input High Voltage</b> I/O ports: with TTL buffer	2.0	—	V <sub>DD</sub>	V	$4.5\text{V} \leq V_{DD} \leq 5.5\text{V}$
D040A			$0.25 V_{DD} + 0.8$	—	V <sub>DD</sub>	V	$2.0\text{V} \leq V_{DD} \leq 4.5\text{V}$
D041		with Schmitt Trigger buffer	$0.8 V_{DD}$	—	V <sub>DD</sub>	V	$2.0\text{V} \leq V_{DD} \leq 5.5\text{V}$
D042		MCLR	$0.8 V_{DD}$	—	V <sub>DD</sub>	V	
D043		OSC1 (XT and LP modes)	1.6	—	V <sub>DD</sub>	V	
D043A		OSC1 (HS mode)	$0.7 V_{DD}$	—	V <sub>DD</sub>	V	
D043B		OSC1 (RC mode)	$0.9 V_{DD}$	—	V <sub>DD</sub>	V	<b>(Note 1)</b>
D060	I <sub>IL</sub>	<b>Input Leakage Current</b> <sup>(2)</sup> I/O ports	—	$\pm 0.1$	$\pm 1$	$\mu\text{A}$	$V_{SS} \leq V_{PIN} \leq V_{DD}$ , Pin at high-impedance
D061		MCLR <sup>(3)</sup>	—	$\pm 0.1$	$\pm 5$	$\mu\text{A}$	$V_{SS} \leq V_{PIN} \leq V_{DD}$
D063		OSC1	—	$\pm 0.1$	$\pm 5$	$\mu\text{A}$	$V_{SS} \leq V_{PIN} \leq V_{DD}$ , XT, HS and LP oscillator configuration
D070*	I <sub>PUR</sub>	GPIO Weak Pull-up Current	50	250	400	$\mu\text{A}$	$V_{DD} = 5.0\text{V}$ , $V_{PIN} = V_{SS}$
D080	V <sub>OL</sub>	<b>Output Low Voltage</b> <sup>(5)</sup> I/O ports	—	—	0.6	V	I <sub>OL</sub> = 8.5 mA, $V_{DD} = 4.5\text{V}$ (Ind.)
D090	V <sub>OH</sub>	<b>Output High Voltage</b> <sup>(5)</sup> I/O ports	$V_{DD} - 0.7$	—	—	V	I <sub>OH</sub> = -3.0 mA, $V_{DD} = 4.5\text{V}$ (Ind.)

\* These parameters are characterized but not tested.

† Data in "Typ" column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** In RC oscillator configuration, the OSC1/CLKIN pin is a Schmitt Trigger input. It is not recommended to use an external clock in RC mode.

**2:** Negative current is defined as current sourced by the pin.

**3:** The leakage current on the MCLR pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.

**4:** See **Section 10.4.1 "Using the Data EEPROM"** for additional information.

**5:** Including OSC2 in CLKOUT mode.

# PIC12F683

## 15.5 DC Characteristics: PIC12F683-I (Industrial) PIC12F683-E (Extended) (Continued)

DC CHARACTERISTICS			Standard Operating Conditions (unless otherwise stated)				
			Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended				
Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
D100	IULP	Ultra Low-Power Wake-Up Current	—	200	—	nA	See Application Note AN879, “Using the Microchip Ultra Low-Power Wake-up Module” (DS00879)
D101*	COSC2	<b>Capacitive Loading Specs on Output Pins</b> OSC2 pin	—	—	15	pF	In XT, HS and LP modes when external clock is used to drive OSC1
D101A*	CIO	All I/O pins	—	—	50	pF	
D120	ED	<b>Data EEPROM Memory</b> Byte Endurance	100K	1M	—	E/W	-40°C ≤ TA ≤ +85°C
D120A	ED	Byte Endurance	10K	100K	—	E/W	+85°C ≤ TA ≤ +125°C
D121	VDRW	VDD for Read/Write	VMIN	—	5.5	V	Using EECON1 to read/write VMIN = Minimum operating voltage
D122	TDEW	Erase/Write Cycle Time	—	5	6	ms	Provided no other specifications are violated
D123	TRETD	Characteristic Retention	40	—	—	Year	
D124	TREF	Number of Total Erase/Write Cycles before Refresh <sup>(4)</sup>	1M	10M	—	E/W	
D130	EP	<b>Program Flash Memory</b> Cell Endurance	10K	100K	—	E/W	-40°C ≤ TA ≤ +85°C
D130A	ED	Cell Endurance	1K	10K	—	E/W	+85°C ≤ TA ≤ +125°C
D131	VPR	VDD for Read	VMIN	—	5.5	V	VMIN = Minimum operating voltage
D132	VPEW	VDD for Erase/Write	4.5	—	5.5	V	Provided no other specifications are violated
D133	TPEW	Erase/Write cycle time	—	2	2.5	ms	
D134	TRETD	Characteristic Retention	40	—	—	Year	

\* These parameters are characterized but not tested.

† Data in "Typ" column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

- Note 1:** In RC oscillator configuration, the OSC1/CLKIN pin is a Schmitt Trigger input. It is not recommended to use an external clock in RC mode.
- 2:** Negative current is defined as current sourced by the pin.
- 3:** The leakage current on the MCLR pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.
- 4:** See **Section 10.4.1 "Using the Data EEPROM"** for additional information.
- 5:** Including OSC2 in CLKOUT mode.

## 15.6 Thermal Considerations

Standard Operating Conditions (unless otherwise stated)					
Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$					
Param No.	Sym	Characteristic	Typ	Units	Conditions
TH01	$\theta_{JA}$	Thermal Resistance Junction to Ambient	84.6	$^{\circ}\text{C}/\text{W}$	8-pin PDIP package
			163.0	$^{\circ}\text{C}/\text{W}$	8-pin SOIC package
			52.4	$^{\circ}\text{C}/\text{W}$	8-pin DFN-S 4x4x0.9 mm package
			46.3	$^{\circ}\text{C}/\text{W}$	8-pin DFN-S 6x5 mm package
TH02	$\theta_{JC}$	Thermal Resistance Junction to Case	41.2	$^{\circ}\text{C}/\text{W}$	8-pin PDIP package
			38.8	$^{\circ}\text{C}/\text{W}$	8-pin SOIC package
			3.0	$^{\circ}\text{C}/\text{W}$	8-pin DFN-S 4x4x0.9 mm package
			2.6	$^{\circ}\text{C}/\text{W}$	8-pin DFN-S 6x5 mm package
TH03	$T_J$	Junction Temperature	150	$^{\circ}\text{C}$	For derated power calculations
TH04	PD	Power Dissipation	—	W	$PD = P_{INTERNAL} + P_{I/O}$
TH05	$P_{INTERNAL}$	Internal Power Dissipation	—	W	$P_{INTERNAL} = I_{DD} \times V_{DD}$ <b>(NOTE 1)</b>
TH06	$P_{I/O}$	I/O Power Dissipation	—	W	$P_{I/O} = \Sigma (I_{OL} \times V_{OL}) + \Sigma (I_{OH} \times (V_{DD} - V_{OH}))$
TH07	$P_{DER}$	Derated Power	—	W	$P_{DER} = (T_J - T_A)/\theta_{JA}$ <b>(NOTE 2, 3)</b>

**Note 1:**  $I_{DD}$  is current to run the chip alone without driving any load on the output pins.

**2:**  $T_A$  = Ambient Temperature.

**3:** Maximum allowable power dissipation is the lower value of either the absolute maximum total power dissipation or derated power ( $P_{DER}$ ).

15.7 Timing Parameter Symbolology

The timing parameter symbols have been created with one of the following formats:

- 1. TppS2ppS
- 2. TppS

<b>T</b>			
F	Frequency	T	Time

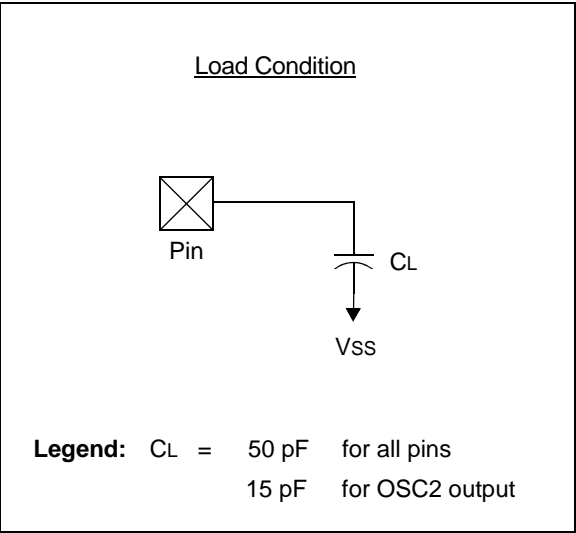
Lowercase letters (pp) and their meanings:

<b>pp</b>			
cc	CCP1	osc	OSC1
ck	CLKOUT	rd	$\overline{RD}$
cs	$\overline{CS}$	rw	$\overline{RD}$ or $\overline{WR}$
di	SDI	sc	SCK
do	SDO	ss	$\overline{SS}$
dt	Data in	t0	T0CKI
io	I/O PORT	t1	T1CKI
mc	$\overline{MCLR}$	wr	$\overline{WR}$

Uppercase letters and their meanings:

<b>S</b>			
F	Fall	P	Period
H	High	R	Rise
I	Invalid (High-impedance)	V	Valid
L	Low	Z	High-impedance

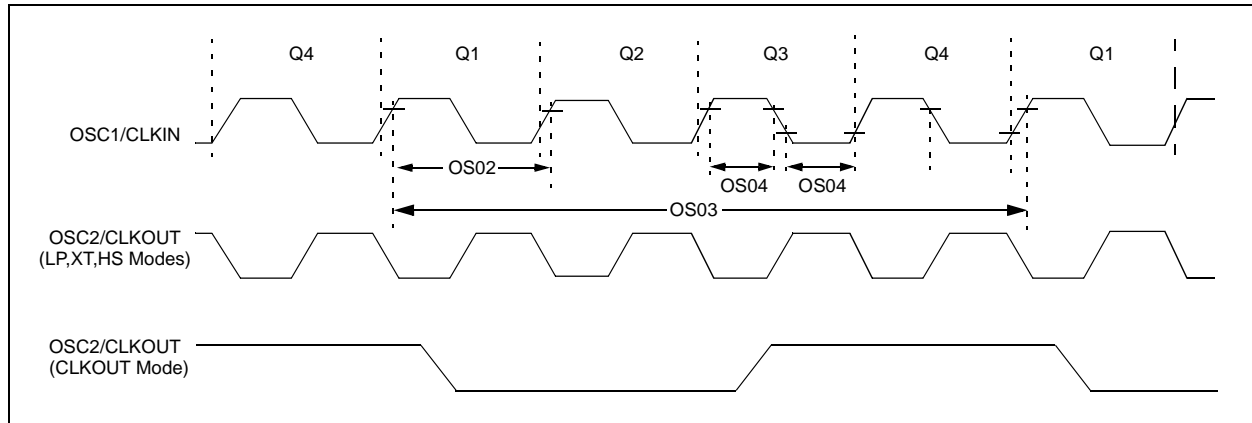
FIGURE 15-3: LOAD CONDITIONS





## 15.8 AC Characteristics: PIC12F683 (Industrial, Extended)

**FIGURE 15-4: CLOCK TIMING**



**TABLE 15-1: CLOCK OSCILLATOR TIMING REQUIREMENTS**

Standard Operating Conditions (unless otherwise stated)							
Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$							
Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
OS01	Fosc	External CLKIN Frequency <sup>(1)</sup>	DC	—	37	kHz	LP Oscillator mode
			DC	—	4	MHz	XT Oscillator mode
			DC	—	20	MHz	HS Oscillator mode
			DC	—	20	MHz	EC Oscillator mode
		Oscillator Frequency <sup>(1)</sup>	—	32.768	—	kHz	LP Oscillator mode
			0.1	—	4	MHz	XT Oscillator mode
			1	—	20	MHz	HS Oscillator mode
			DC	—	4	MHz	RC Oscillator mode
OS02	Tosc	External CLKIN Period <sup>(1)</sup>	27	—	•	μs	LP Oscillator mode
			250	—	•	ns	XT Oscillator mode
			50	—	•	ns	HS Oscillator mode
			50	—	•	ns	EC Oscillator mode
		Oscillator Period <sup>(1)</sup>	—	30.5	—	μs	LP Oscillator mode
			250	—	10,000	ns	XT Oscillator mode
			50	—	1,000	ns	HS Oscillator mode
			250	—	—	ns	RC Oscillator mode
OS03	Tcy	Instruction Cycle Time <sup>(1)</sup>	200	Tcy	DC	ns	Tcy = 4/Fosc
OS04*	TosH, TosL	External CLKIN High, External CLKIN Low	2	—	—	μs	LP oscillator
			100	—	—	ns	XT oscillator
			20	—	—	ns	HS oscillator
OS05*	TosR, TosF	External CLKIN Rise, External CLKIN Fall	0	—	•	ns	LP oscillator
			0	—	•	ns	XT oscillator
			0	—	•	ns	HS oscillator

\* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** Instruction cycle period (Tcy) equals four times the input oscillator time base period. All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at "min" values with an external clock applied to OSC1 pin. When an external clock input is used, the "max" cycle time limit is "DC" (no clock) for all devices.

# PIC12F683

**TABLE 15-2: OSCILLATOR PARAMETERS**

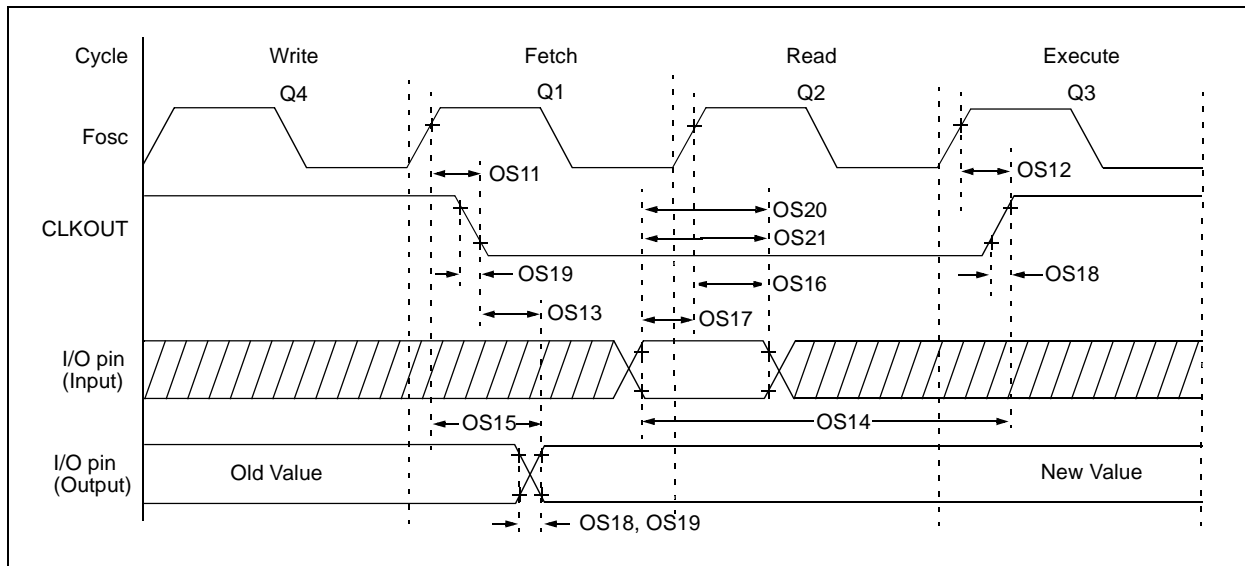
Standard Operating Conditions (unless otherwise stated)								
Operating Temperature $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$								
Param No.	Sym	Characteristic	Freq. Tolerance	Min	Typ†	Max	Units	Conditions
OS06	TWARM	Internal Oscillator Switch when running <sup>(3)</sup>	—	—	—	2	Tosc	Slowest clock
OS07	Tsc	Fail-Safe Sample Clock Period <sup>(1)</sup>	—	—	21	—	ms	LFINTOSC/64
OS08	HFosc	Internal Calibrated HFINTOSC Frequency <sup>(2)</sup>	$\pm 1\%$	7.92	8.0	8.08	MHz	$V_{DD} = 3.5\text{V}$ , $25^{\circ}\text{C}$
			$\pm 2\%$	7.84	8.0	8.16	MHz	$2.5\text{V} \leq V_{DD} \leq 5.5\text{V}$ , $0^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$
			$\pm 5\%$	7.60	8.0	8.40	MHz	$2.0\text{V} \leq V_{DD} \leq 5.5\text{V}$ , $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ (Ind.), $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ (Ext.)
OS09*	LFosc	Internal Uncalibrated LFINTOSC Frequency	—	15	31	45	kHz	
OS10*	Tiosc ST	HFINTOSC Oscillator Wake-up from Sleep Start-up Time	—	5.5	12	24	$\mu\text{s}$	$V_{DD} = 2.0\text{V}$ , $-40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$
			—	3.5	7	14	$\mu\text{s}$	$V_{DD} = 3.0\text{V}$ , $-40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$
			—	3	6	11	$\mu\text{s}$	$V_{DD} = 5.0\text{V}$ , $-40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$

\* These parameters are characterized but not tested.

† Data in "Typ" column is at 5.0V,  $25^{\circ}\text{C}$  unless otherwise stated. These parameters are for design guidance only and are not tested.

- Note 1:** Instruction cycle period ( $T_{CY}$ ) equals four times the input oscillator time base period. All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at "min" values with an external clock applied to the OSC1 pin. When an external clock input is used, the "max" cycle time limit is "DC" (no clock) for all devices.
- 2:** To ensure these oscillator frequency tolerances,  $V_{DD}$  and  $V_{SS}$  must be capacitively decoupled as close to the device as possible. 0.1  $\mu\text{F}$  and 0.01  $\mu\text{F}$  values in parallel are recommended.
- 3:** By design.

**FIGURE 15-5: CLKOUT AND I/O TIMING**



**TABLE 15-3: CLKOUT AND I/O TIMING PARAMETERS**

Standard Operating Conditions (unless otherwise stated)							
Operating Temperature $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$							
Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
OS11	TosH2ckL	Fosc↑ to CLKOUT↓ <sup>(1)</sup>	—	—	70	ns	VDD = 5.0V
OS12	TosH2ckH	Fosc↑ to CLKOUT↑ <sup>(1)</sup>	—	—	72	ns	VDD = 5.0V
OS13	TckL2ioV	CLKOUT↓ to Port out valid <sup>(1)</sup>	—	—	20	ns	
OS14	TioV2ckH	Port input valid before CLKOUT↑ <sup>(1)</sup>	Tosc + 200 ns	—	—	ns	
OS15*	TosH2ioV	Fosc↑ (Q1 cycle) to Port out valid	—	50	70	ns	VDD = 5.0V
OS16	TosH2ioI	Fosc↑ (Q2 cycle) to Port input invalid (I/O in hold time)	50	—	—	ns	VDD = 5.0V
OS17	TioV2osH	Port input valid to Fosc↑ (Q2 cycle) (I/O in setup time)	20	—	—	ns	
OS18	TioR	Port output rise time <sup>(2)</sup>	—	15 40	72 32	ns	VDD = 2.0V VDD = 5.0V
OS19	TioF	Port output fall time <sup>(2)</sup>	—	28 15	55 30	ns	VDD = 2.0V VDD = 5.0V
OS20*	TINP	INT pin input high or low time	25	—	—	ns	
OS21*	TGPP	GPIO interrupt-on-change new input level time	TCY	—	—	ns	

\* These parameters are characterized but not tested.

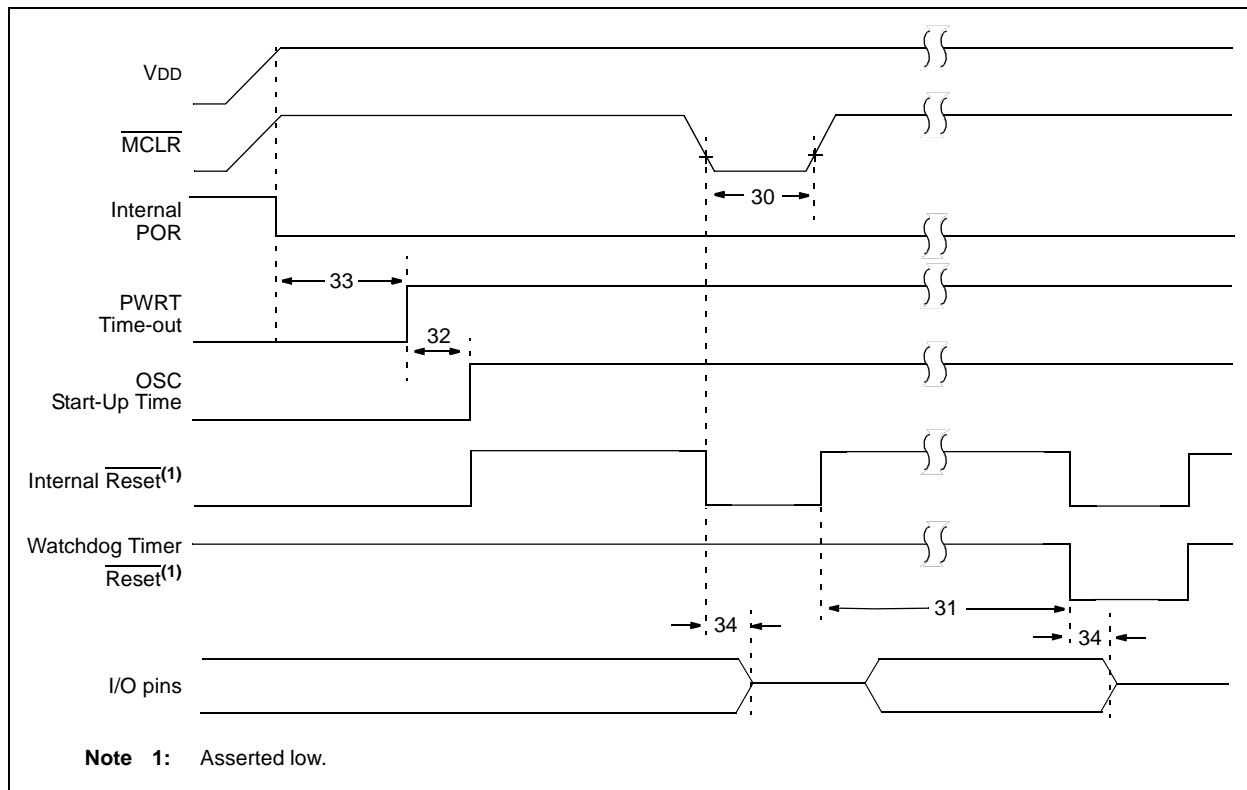
† Data in "Typ" column is at 5.0V, 25°C unless otherwise stated.

**Note 1:** Measurements are taken in RC mode where CLKOUT output is 4 x TOSC.

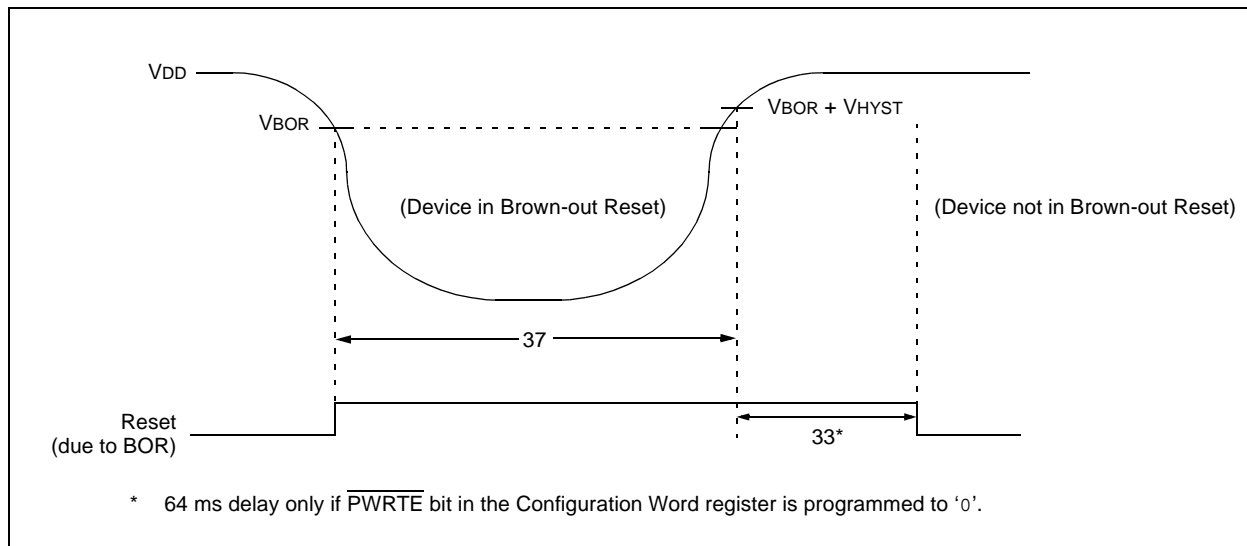
**2:** Includes OSC2 in CLKOUT mode.

# PIC12F683

**FIGURE 15-6: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING**



**FIGURE 15-7: BROWN-OUT RESET TIMING AND CHARACTERISTICS**



**TABLE 15-4: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER, POWER-UP TIMER AND BROWN-OUT RESET PARAMETERS**

Standard Operating Conditions (unless otherwise stated)							
Operating Temperature $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$							
Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
30	TMCL	MCLR Pulse Width (low)	2 5	— —	— —	$\mu\text{s}$ $\mu\text{s}$	$V_{DD} = 5\text{V}$ , $-40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$ $V_{DD} = 5\text{V}$
31	TWDT	Watchdog Timer Time-out Period (No Prescaler)	10 10	16 16	29 31	ms ms	$V_{DD} = 5\text{V}$ , $-40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$ $V_{DD} = 5\text{V}$
32	TOST	Oscillation Start-up Timer Period <sup>(1, 2)</sup>	—	1024	—	T <sub>OSC</sub>	(NOTE 3)
33*	TPWRT	Power-up Timer Period	40	65	140	ms	
34*	TIOZ	I/O High-impedance from MCLR Low or Watchdog Timer Reset	—	—	2.0	$\mu\text{s}$	
35	VBOR	Brown-out Reset Voltage	2.0	—	2.2	V	(NOTE 4)
36*	VHYST	Brown-out Reset Hysteresis	—	50	—	mV	
37*	TBOR	Brown-out Reset Minimum Detection Period	100	—	—	$\mu\text{s}$	$V_{DD} \leq V_{BOR}$

\* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** Instruction cycle period (T<sub>CY</sub>) equals four times the input oscillator time base period. All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at "min" values with an external clock applied to the OSC1 pin. When an external clock input is used, the "max" cycle time limit is "DC" (no clock) for all devices.

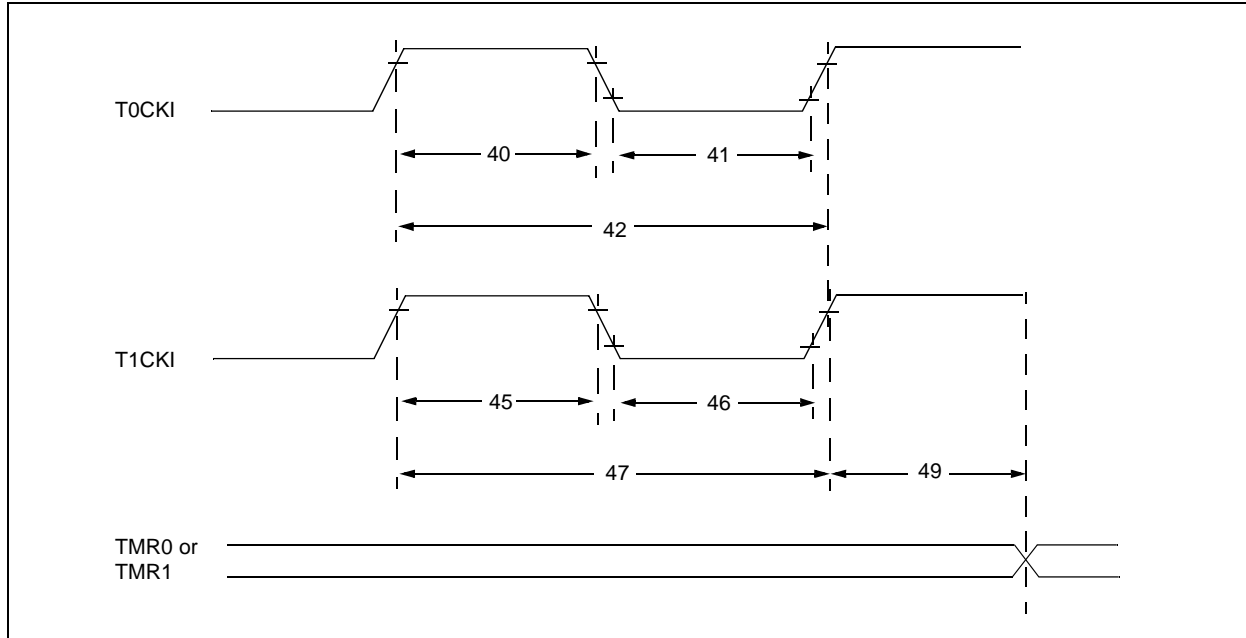
**2:** By design.

**3:** Period of the slower clock.

**4:** To ensure these voltage tolerances, V<sub>DD</sub> and V<sub>SS</sub> must be capacitively decoupled as close to the device as possible. 0.1  $\mu\text{F}$  and 0.01  $\mu\text{F}$  values in parallel are recommended.

# PIC12F683

**FIGURE 15-8: TIMER0 AND TIMER1 EXTERNAL CLOCK TIMINGS**



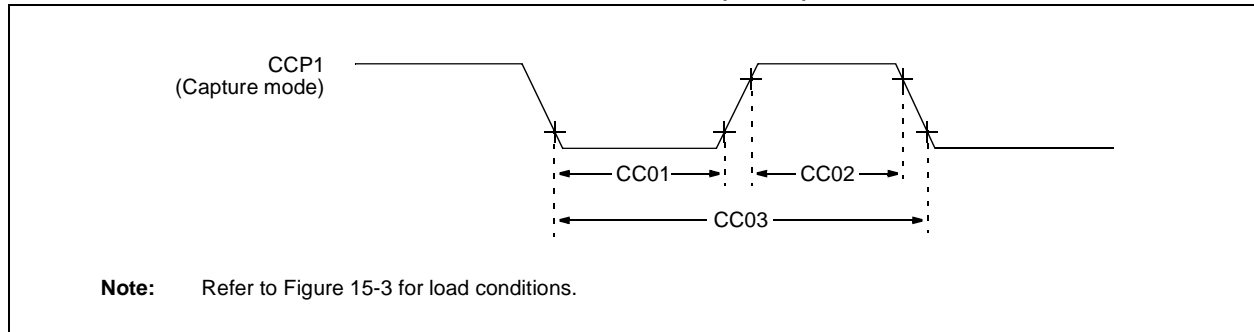
**TABLE 15-5: TIMER0 AND TIMER1 EXTERNAL CLOCK REQUIREMENTS**

Standard Operating Conditions (unless otherwise stated)								
Operating Temperature $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$								
Param No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions
40*	Tt0H	T0CKI High Pulse Width	No Prescaler	$0.5 T_{CY} + 20$	—	—	ns	
			With Prescaler	10	—	—	ns	
41*	Tt0L	T0CKI Low Pulse Width	No Prescaler	$0.5 T_{CY} + 20$	—	—	ns	
			With Prescaler	10	—	—	ns	
42*	Tt0P	T0CKI Period		Greater of: $20$ or $\frac{T_{CY} + 40}{N}$	—	—	ns	N = prescale value (2, 4, ..., 256)
45*	Tt1H	T1CKI High Time	Synchronous, No Prescaler	$0.5 T_{CY} + 20$	—	—	ns	
			Synchronous, with Prescaler	15	—	—	ns	
			Asynchronous	30	—	—	ns	
46*	Tt1L	T1CKI Low Time	Synchronous, No Prescaler	$0.5 T_{CY} + 20$	—	—	ns	
			Synchronous, with Prescaler	15	—	—	ns	
			Asynchronous	30	—	—	ns	
47*	Tt1P	T1CKI Input Period	Synchronous	Greater of: $30$ or $\frac{T_{CY} + 40}{N}$	—	—	ns	N = prescale value (1, 2, 4, 8)
			Asynchronous	60	—	—	ns	
48	Ft1	Timer1 Oscillator Input Frequency Range (oscillator enabled by setting bit T1OSCEN)		—	32.768	—	kHz	
49*	TCKEZTMR1	Delay from External Clock Edge to Timer Increment		$2 T_{osc}$	—	$7 T_{osc}$	—	Timers in Sync mode

\* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**FIGURE 15-9: CAPTURE/COMPARE/PWM TIMINGS (ECCP)**



**TABLE 15-6: CAPTURE/COMPARE/PWM REQUIREMENTS (ECCP)**

Standard Operating Conditions (unless otherwise stated)								
Operating Temperature $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$								
Param No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions
CC01*	TccL	CCP1 Input Low Time	No Prescaler	$0.5T_{CY} + 20$	—	—	ns	
			With Prescaler	20	—	—	ns	
CC02*	TccH	CCP1 Input High Time	No Prescaler	$0.5T_{CY} + 20$	—	—	ns	
			With Prescaler	20	—	—	ns	
CC03*	TccP	CCP1 Input Period		$\frac{3T_{CY} + 40}{N}$	—	—	ns	N = prescale value (1, 4 or 16)

\* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

# PIC12F683

**TABLE 15-7: COMPARATOR SPECIFICATIONS**

Standard Operating Conditions (unless otherwise stated)								
Operating Temperature $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$								
Param No.	Sym	Characteristics		Min	Typ†	Max	Units	Comments
CM01	VOS	Input Offset Voltage		—	$\pm 5.0$	$\pm 10$	mV	$(V_{DD} - 1.5)/2$
CM02	VCM	Input Common Mode Voltage		0	—	$V_{DD} - 1.5$	V	
CM03*	CMRR	Common Mode Rejection Ratio		+55	—	—	dB	
CM04*	TRT	Response Time	Falling	—	150	600	ns	<b>(NOTE 1)</b>
			Rising	—	200	1000	ns	
CM05*	TMC2COV	Comparator Mode Change to Output Valid		—	—	10	$\mu\text{s}$	

\* These parameters are characterized but not tested.

† Data in “Typ” column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** Response time is measured with one comparator input at  $(V_{DD} - 1.5)/2 - 100\text{ mV}$  to  $(V_{DD} - 1.5)/2 + 20\text{ mV}$ .

**TABLE 15-8: COMPARATOR VOLTAGE REFERENCE (CVREF) SPECIFICATIONS**

Standard Operating Conditions (unless otherwise stated)							
Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$							
Param No.	Sym	Characteristics	Min	Typ†	Max	Units	Comments
CV01*	CLSB	Step Size <sup>(2)</sup>	—	$V_{DD}/24$	—	V	Low Range (VRR = 1)
			—	$V_{DD}/32$	—	V	High Range (VRR = 0)
CV02*	CACC	Absolute Accuracy	—	—	$\pm 1/2$	LSb	Low Range (VRR = 1)
			—	—	$\pm 1/2$	LSb	High Range (VRR = 0)
CV03*	CR	Unit Resistor Value (R)	—	2k	—	$\Omega$	
CV04*	CST	Settling Time <sup>(1)</sup>	—	—	10	$\mu\text{s}$	

\* These parameters are characterized but not tested.

† Data in “Typ” column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** Settling time measured while  $VRR = 1$  and  $VR<3:0>$  transitions from ‘0000’ to ‘1111’.

**2:** See **Section 8.11 “Comparator Voltage Reference”** for more information.



**TABLE 15-9: PIC12F683 A/D CONVERTER (ADC) CHARACTERISTICS**

Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$							
Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
AD01	NR	Resolution	—	—	10 bits	bit	
AD02	EIL	Integral Error	—	—	$\pm 1$	LSb	$V_{\text{REF}} = 5.12\text{V}$
AD03	EDL	Differential Error	—	—	$\pm 1$	LSb	No missing codes to 10 bits $V_{\text{REF}} = 5.12\text{V}$
AD04	EOFF	Offset Error	—	—	$\pm 1$	LSb	$V_{\text{REF}} = 5.12\text{V}$
AD07	EGN	Gain Error	—	—	$\pm 1$	LSb	$V_{\text{REF}} = 5.12\text{V}$
AD06 AD06A	VREF	Reference Voltage <sup>(3)</sup>	2.2 2.7	—	— VDD	V	Absolute minimum to ensure 1 LSb accuracy
AD07	VAIN	Full-Scale Range	VSS	—	VREF	V	
AD08	ZAIN	Recommended Impedance of Analog Voltage Source	—	—	10	k $\Omega$	
AD09*	IREF	VREF Input Current <sup>(3)</sup>	10	—	1000	$\mu\text{A}$	During VAIN acquisition. Based on differential of VHOLD to VAIN.
			—	—	50	$\mu\text{A}$	During A/D conversion cycle.

\* These parameters are characterized but not tested.

† Data in "Typ" column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** Total Absolute Error includes integral, differential, offset and gain errors.

**2:** The A/D conversion result never decreases with an increase in the input voltage and has no missing codes.

**3:** ADC VREF is from external VREF or VDD pin, whichever is selected as reference input.

**4:** When ADC is off, it will not consume any current other than leakage current. The power-down current specification includes any such leakage from the ADC module.

# PIC12F683

**TABLE 15-10: PIC12F683 A/D CONVERSION REQUIREMENTS**

Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$							
Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
AD130*	TAD	A/D Clock Period	1.6	—	9.0	$\mu\text{s}$	TOSC-based, $V_{\text{REF}} \geq 3.0\text{V}$
			3.0	—	9.0	$\mu\text{s}$	TOSC-based, $V_{\text{REF}}$ full range
		A/D Internal RC Oscillator Period	3.0	6.0	9.0	$\mu\text{s}$	ADCS<1:0> = 11 (ADRC mode) At $V_{\text{DD}} = 2.5\text{V}$
			1.6	4.0	6.0	$\mu\text{s}$	At $V_{\text{DD}} = 5.0\text{V}$
AD131	TCNV	Conversion Time (not including Acquisition Time) <sup>(1)</sup>	—	11	—	TAD	Set GO/DONE bit to new data in A/D Result register.
AD132*	TACQ	Acquisition Time	—	11.5	—	$\mu\text{s}$	
AD133*	TAMP	Amplifier Settling Time	—	—	5	$\mu\text{s}$	
AD134	TGO	Q4 to A/D Clock Start	—	Tosc/2	—	—	If the A/D clock source is selected as RC, a time of $T_{\text{CY}}$ is added before the A/D clock starts. This allows the SLEEP instruction to be executed.
			—	Tosc/2 + $T_{\text{CY}}$	—	—	

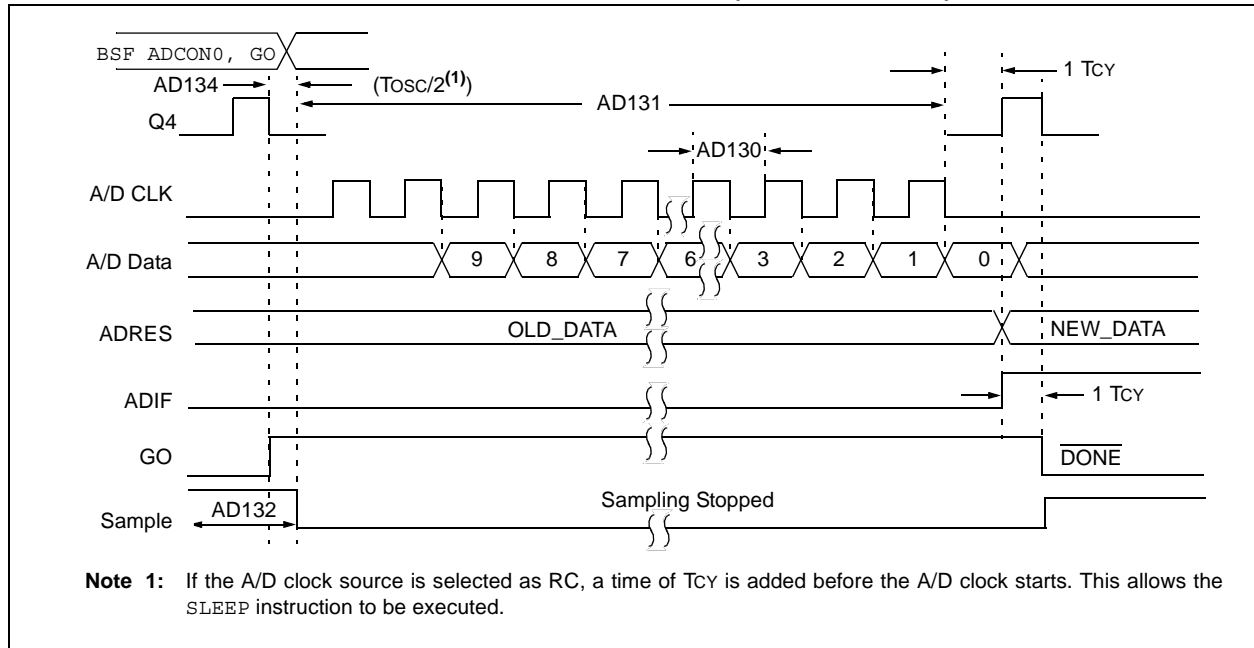
\* These parameters are characterized but not tested.

† Data in "Typ" column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

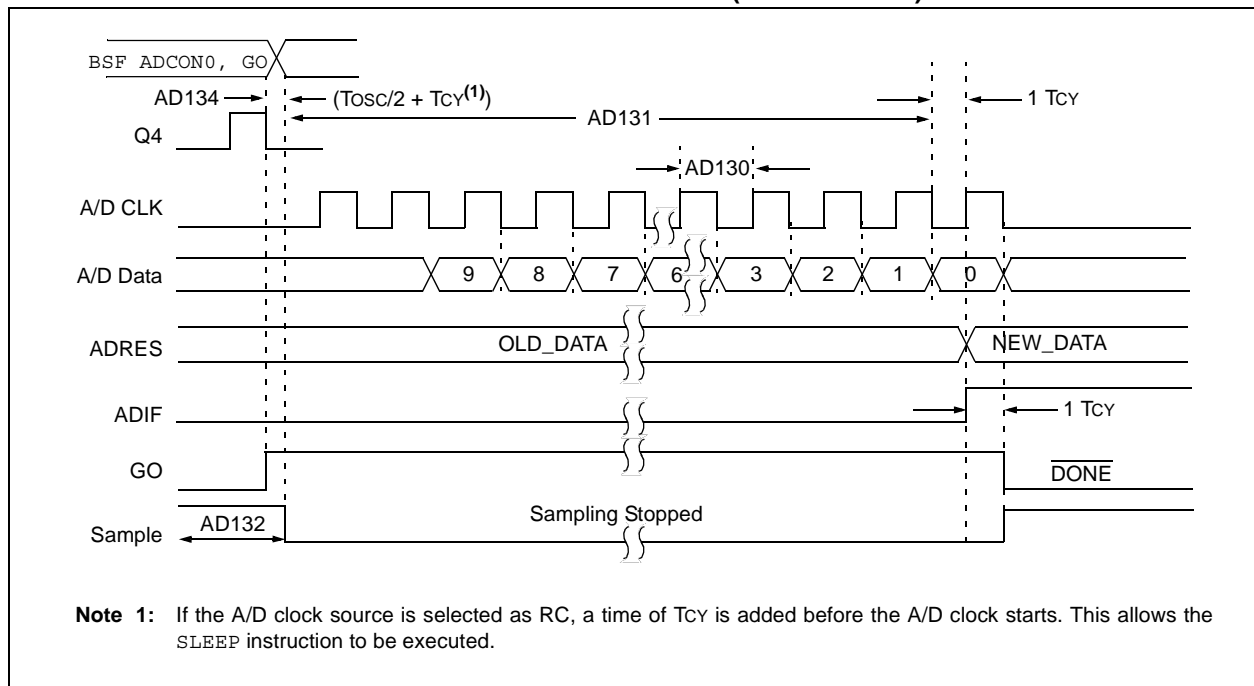
**Note 1:** ADRESH and ADRESL registers may be read on the following  $T_{\text{CY}}$  cycle.

**2:** See Section 9.3 "A/D Acquisition Requirements" for minimum conditions.

**FIGURE 15-10: PIC12F683 A/D CONVERSION TIMING (NORMAL MODE)**



**FIGURE 15-11: PIC12F683 A/D CONVERSION TIMING (SLEEP MODE)**



# PIC12F683

---

NOTES:

## 16.0 DC AND AC CHARACTERISTICS GRAPHS AND TABLES

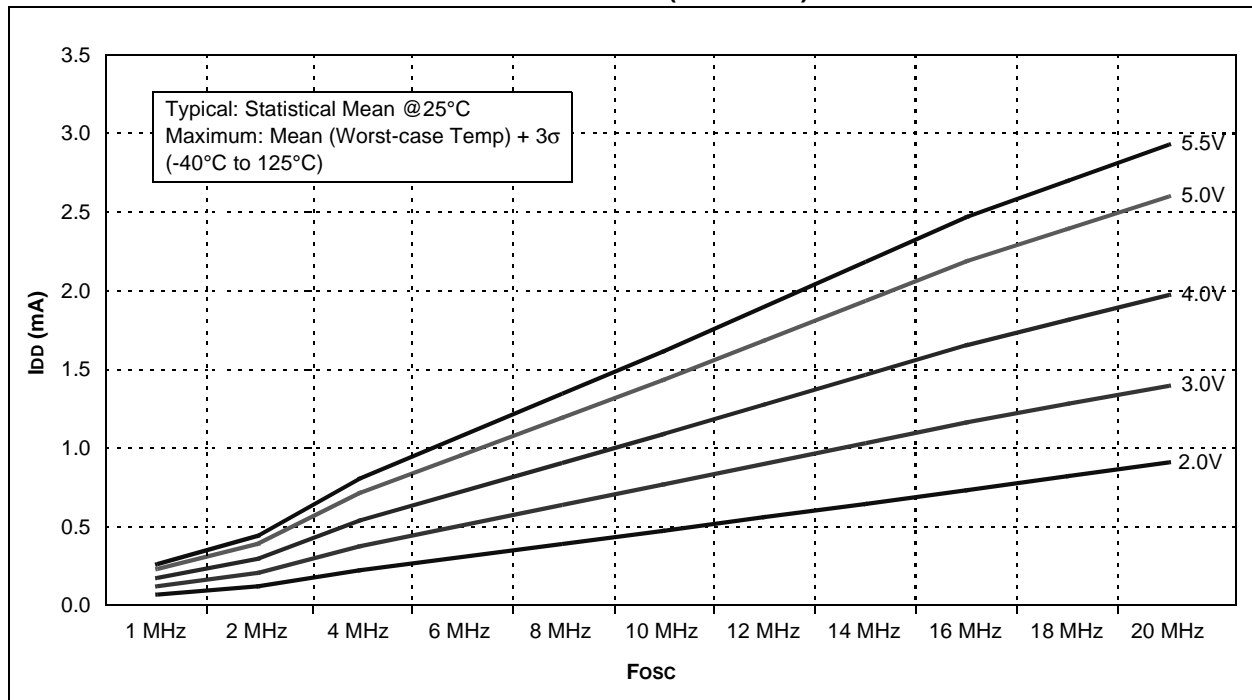
The graphs and tables provided in this section are for **design guidance** and are **not tested**.

In some graphs or tables, the data presented are **outside specified operating range** (i.e., outside specified  $V_{DD}$  range). This is for **information only** and devices are ensured to operate properly only within the specified range.

**Note:** The graphs and tables provided following this note are a statistical summary based on a limited number of samples and are provided for informational purposes only. The performance characteristics listed herein are not tested or guaranteed. In some graphs or tables, the data presented may be outside the specified operating range (e.g., outside specified power supply range) and therefore, outside the warranted range.

“Typical” represents the mean of the distribution at 25°C. “Maximum” or “minimum” represents (mean + 3 $\sigma$ ) or (mean - 3 $\sigma$ ) respectively, where  $\sigma$  is a standard deviation, over each temperature range.

**FIGURE 16-1: TYPICAL  $I_{DD}$  vs.  $F_{osc}$  OVER  $V_{DD}$  (EC MODE)**



# PIC12F683

FIGURE 16-2: MAXIMUM  $I_{DD}$  vs.  $F_{osc}$  OVER  $V_{DD}$  (EC MODE)

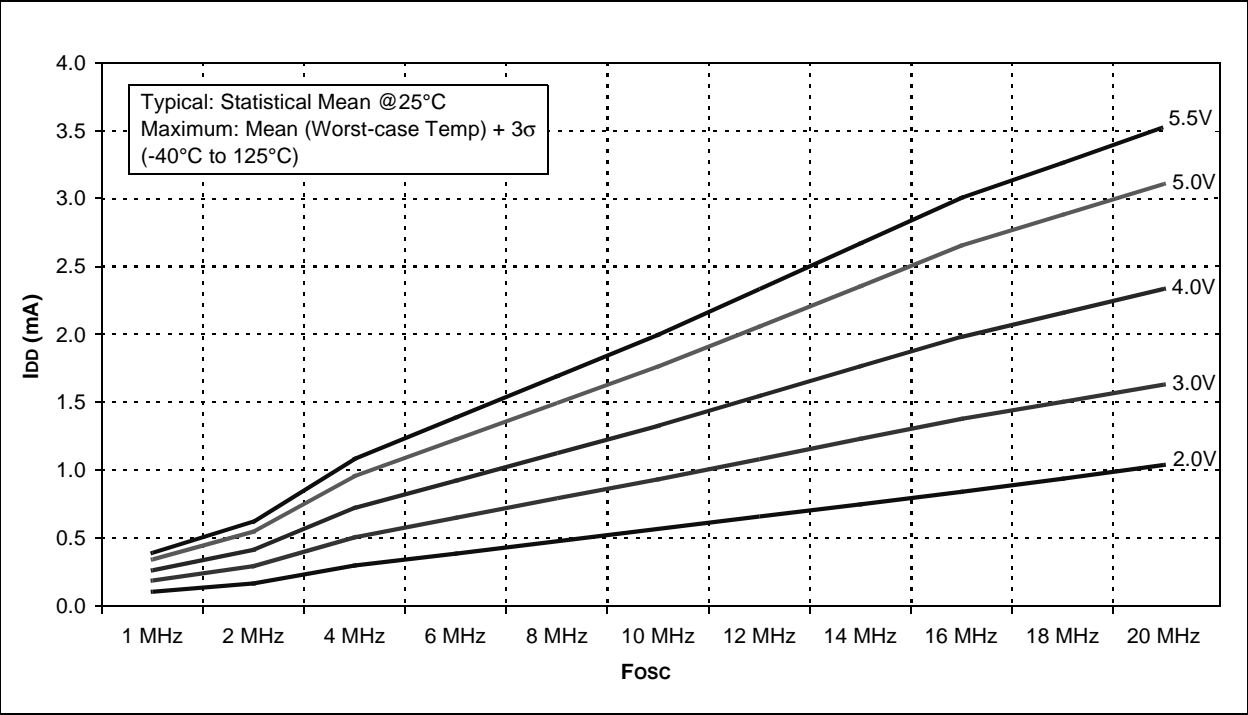


FIGURE 16-3: TYPICAL  $I_{DD}$  vs.  $F_{osc}$  OVER  $V_{DD}$  (HS MODE)

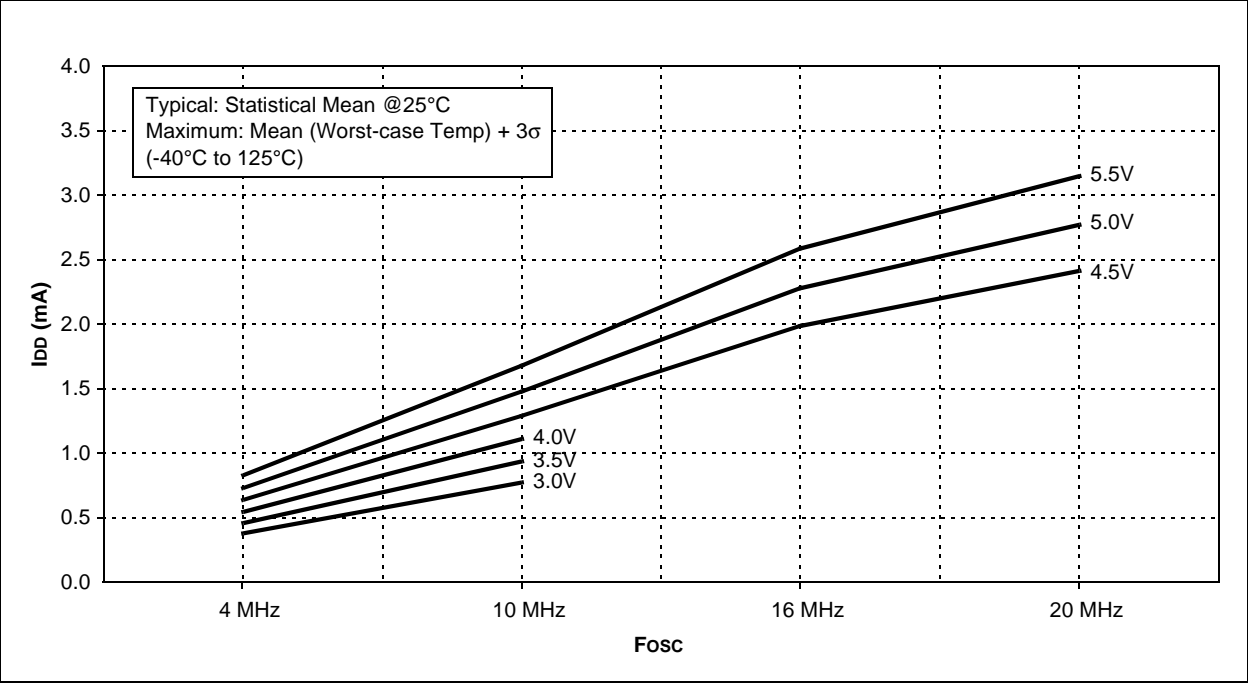


FIGURE 16-4: MAXIMUM I<sub>DD</sub> vs. F<sub>osc</sub> OVER V<sub>DD</sub> (HS MODE)

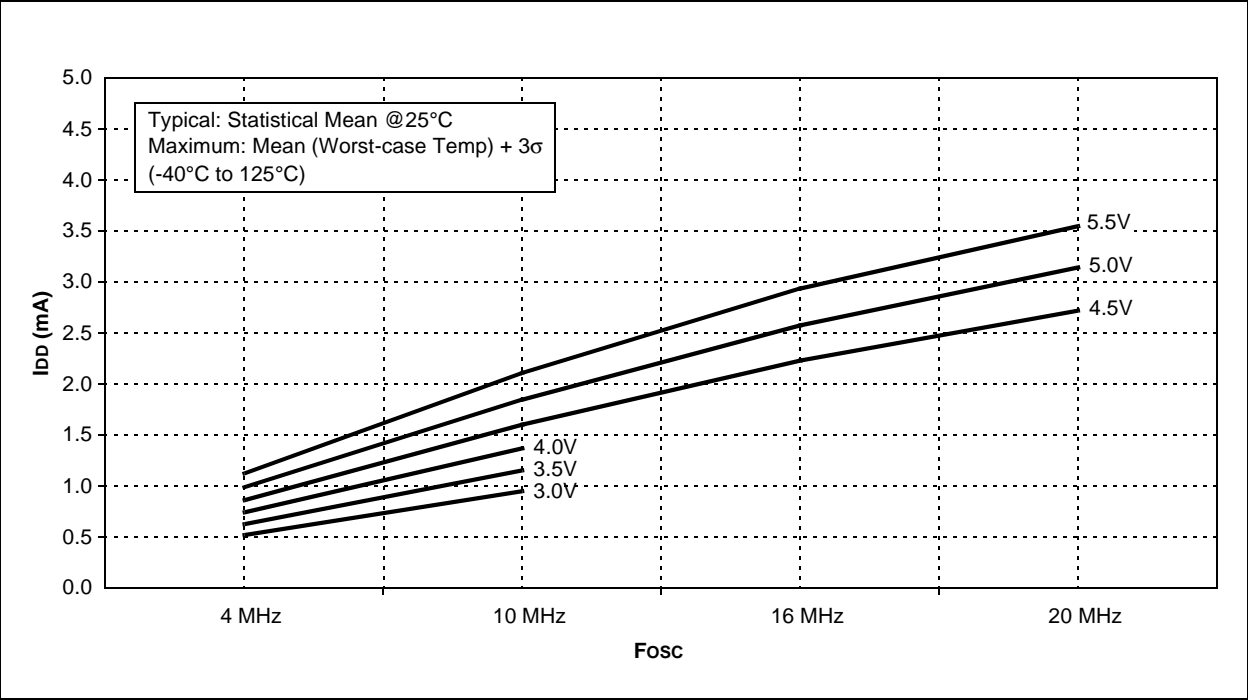
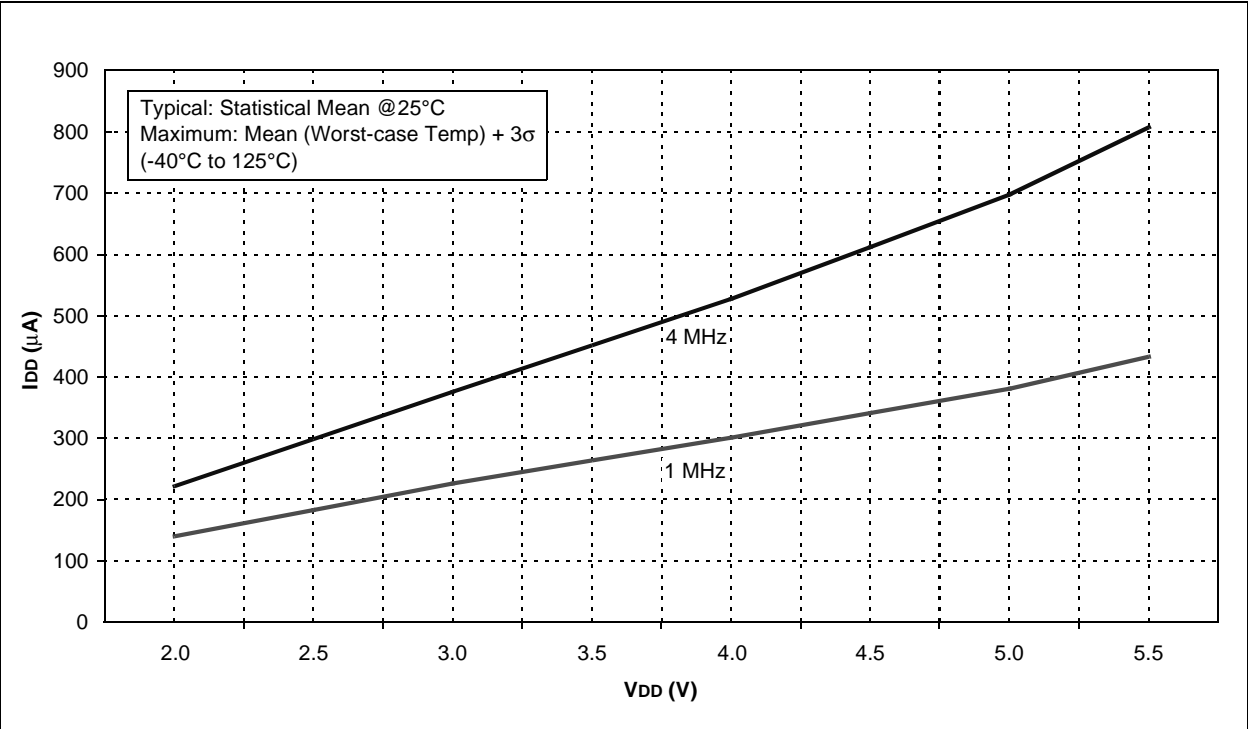


FIGURE 16-5: TYPICAL I<sub>DD</sub> vs. V<sub>DD</sub> OVER F<sub>osc</sub> (XT MODE)



# PIC12F683

FIGURE 16-6: MAXIMUM  $I_{DD}$  vs.  $V_{DD}$  OVER  $F_{osc}$  (XT MODE)

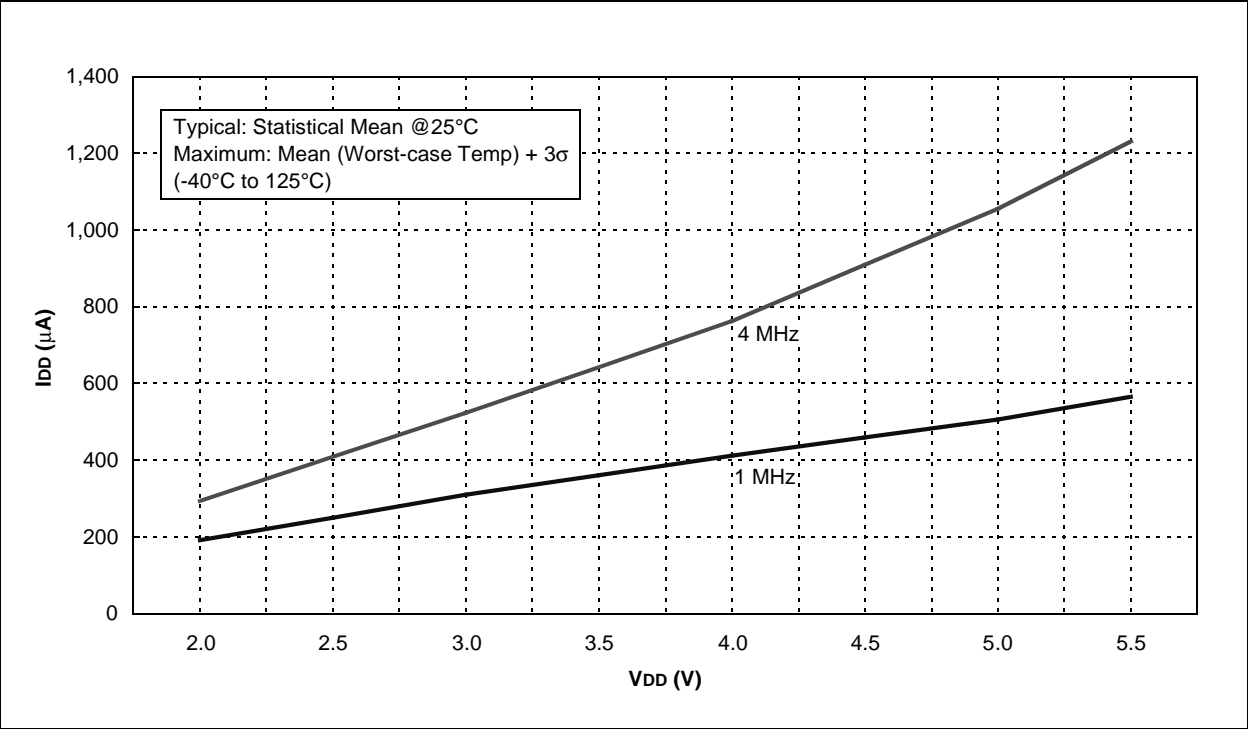
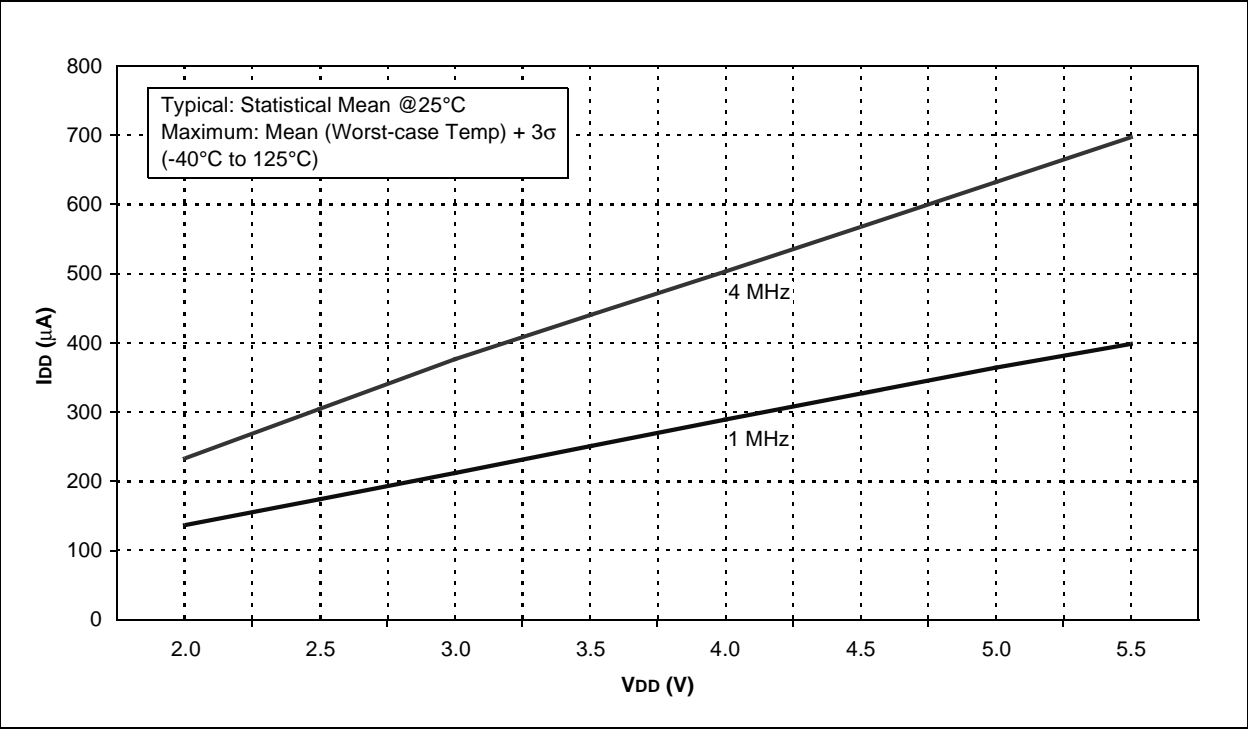
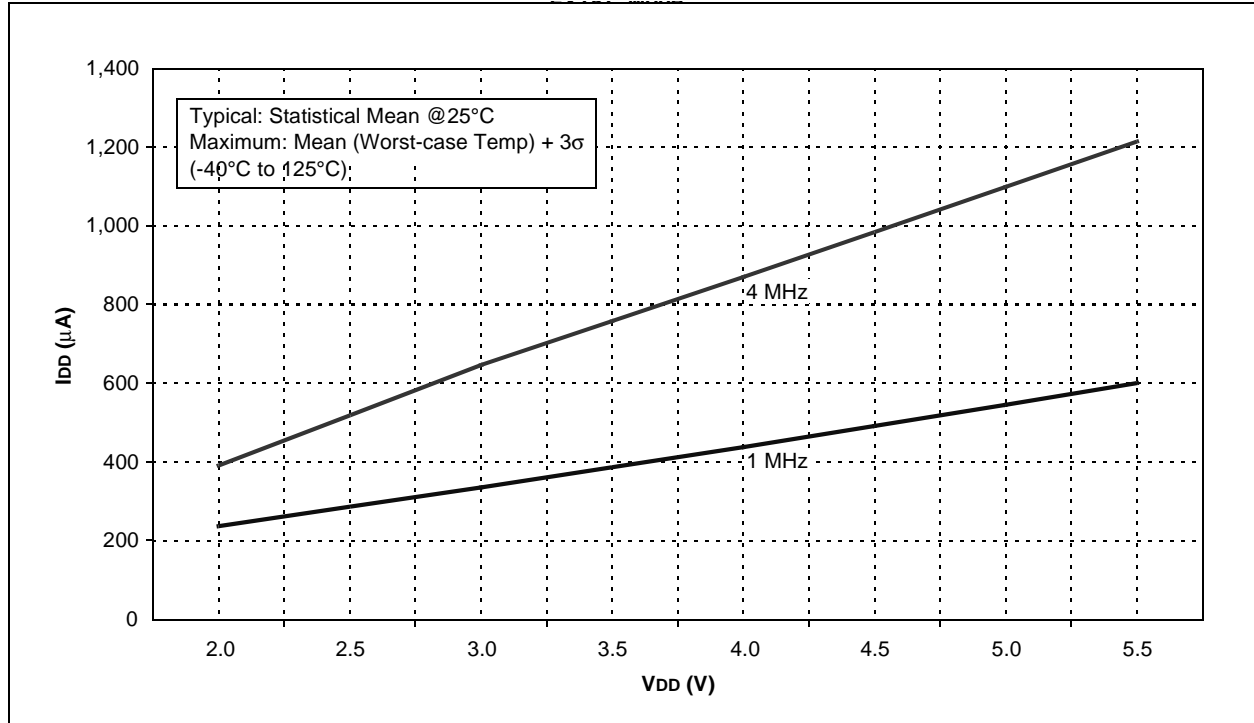


FIGURE 16-7: TYPICAL  $I_{DD}$  vs.  $V_{DD}$  OVER  $F_{osc}$  (EXTRC MODE)

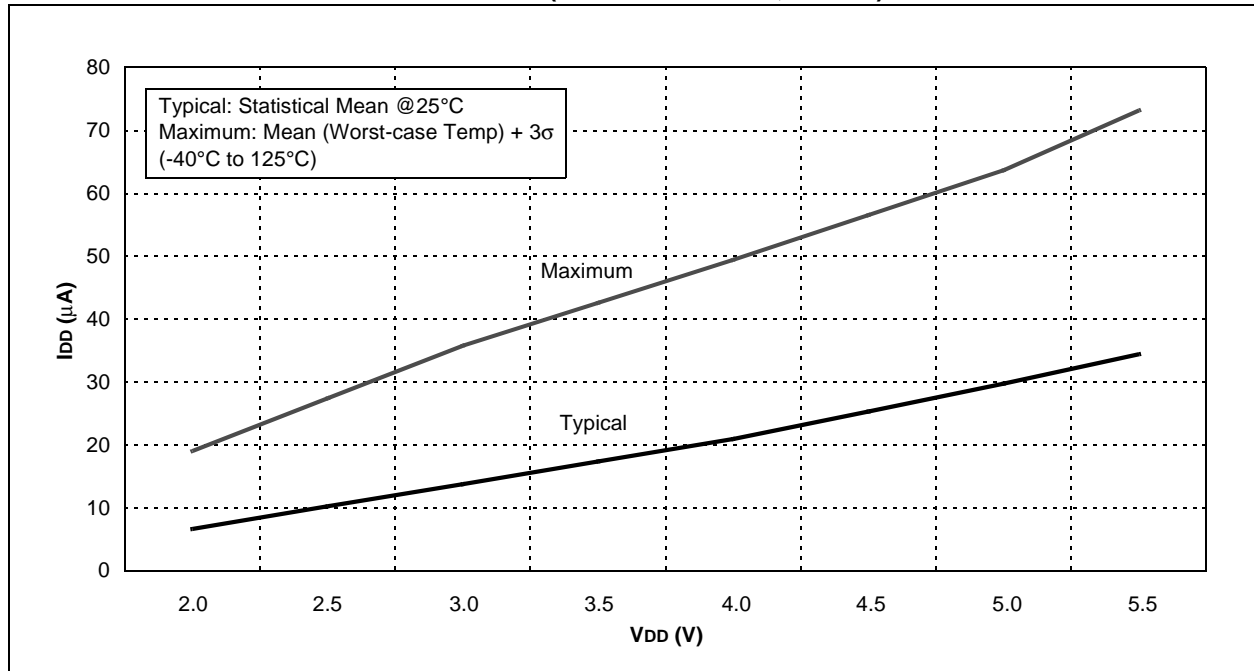




**FIGURE 16-8: MAXIMUM  $I_{DD}$  vs.  $V_{DD}$  (EXTRC MODE)**



**FIGURE 16-9:  $I_{DD}$  vs.  $V_{DD}$  OVER  $F_{OSC}$  (LFINTOSC MODE, 31 kHz)**



# PIC12F683

FIGURE 16-10: I<sub>DD</sub> vs. V<sub>DD</sub> (LP MODE)

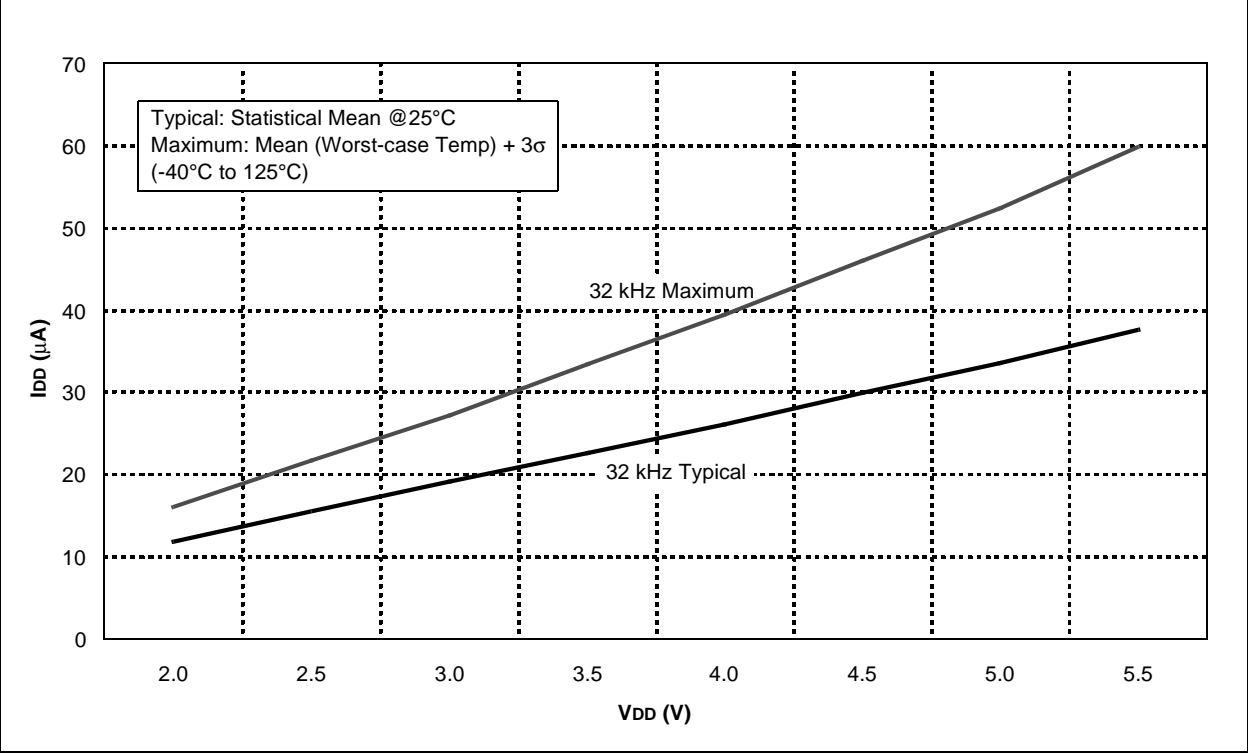
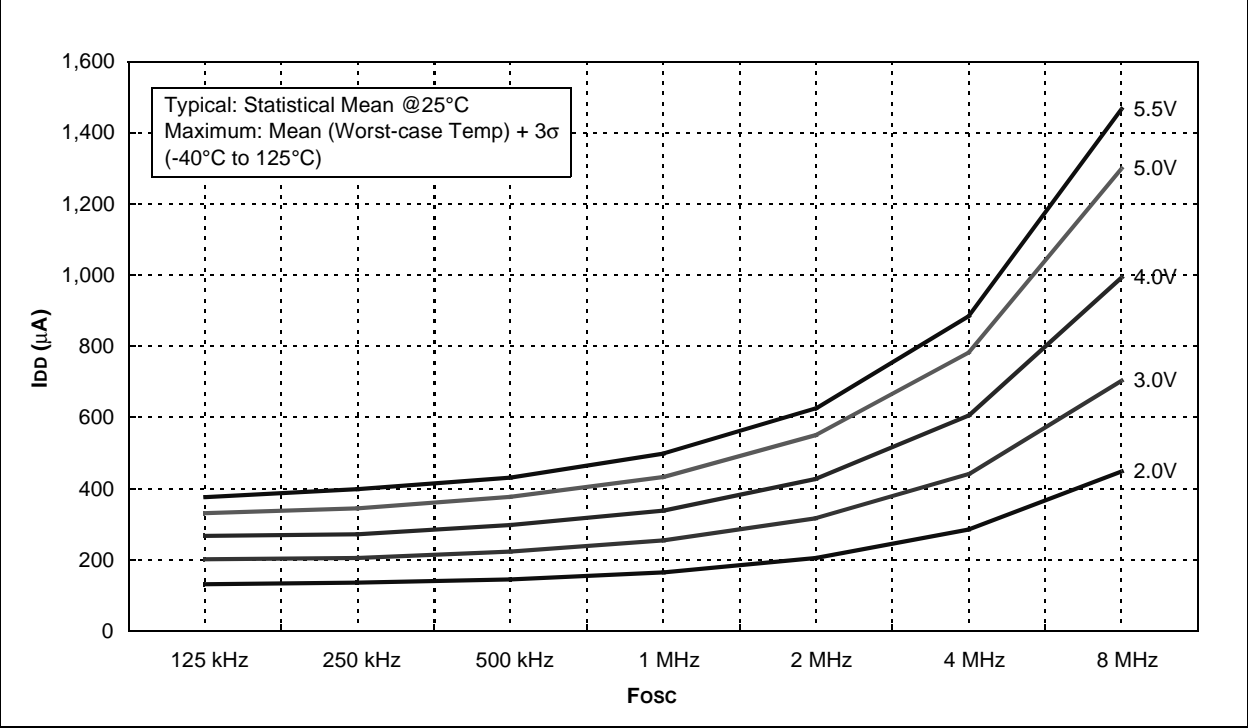
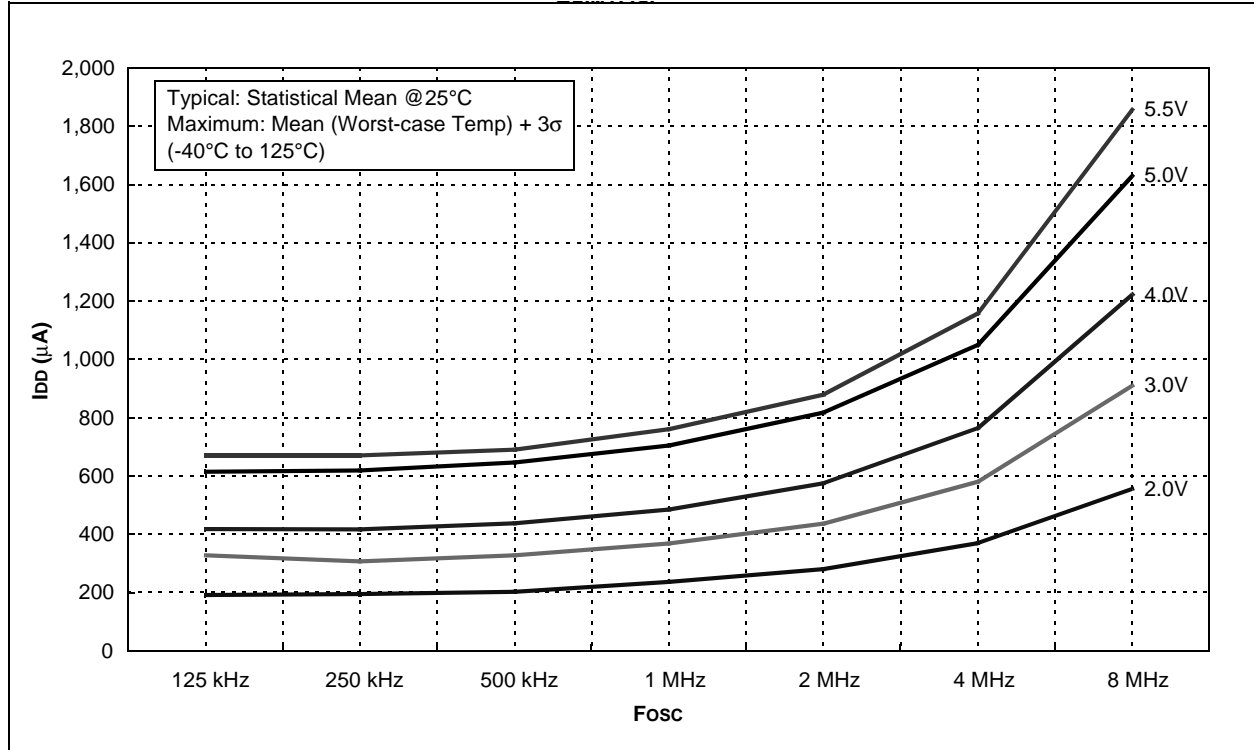


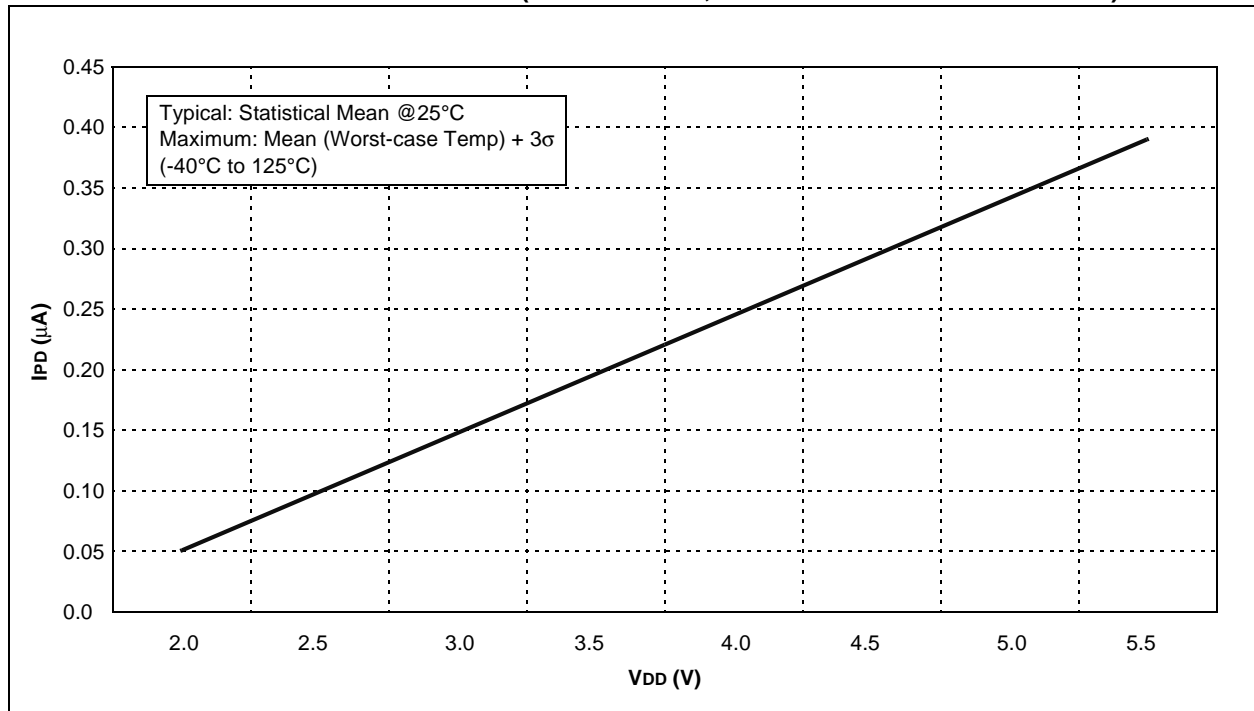
FIGURE 16-11: TYPICAL I<sub>DD</sub> vs. F<sub>osc</sub> OVER V<sub>DD</sub> (HFINTOSC MODE)



**FIGURE 16-12: MAXIMUM  $I_{DD}$  vs.  $F_{osc}$  OVER  $V_{DD}$  (HFINTOSC MODE)**



**FIGURE 16-13: TYPICAL  $I_{PD}$  vs.  $V_{DD}$  (SLEEP MODE, ALL PERIPHERALS DISABLED)**



# PIC12F683

FIGURE 16-14: MAXIMUM  $I_{PD}$  vs.  $V_{DD}$  (SLEEP MODE, ALL PERIPHERALS DISABLED)

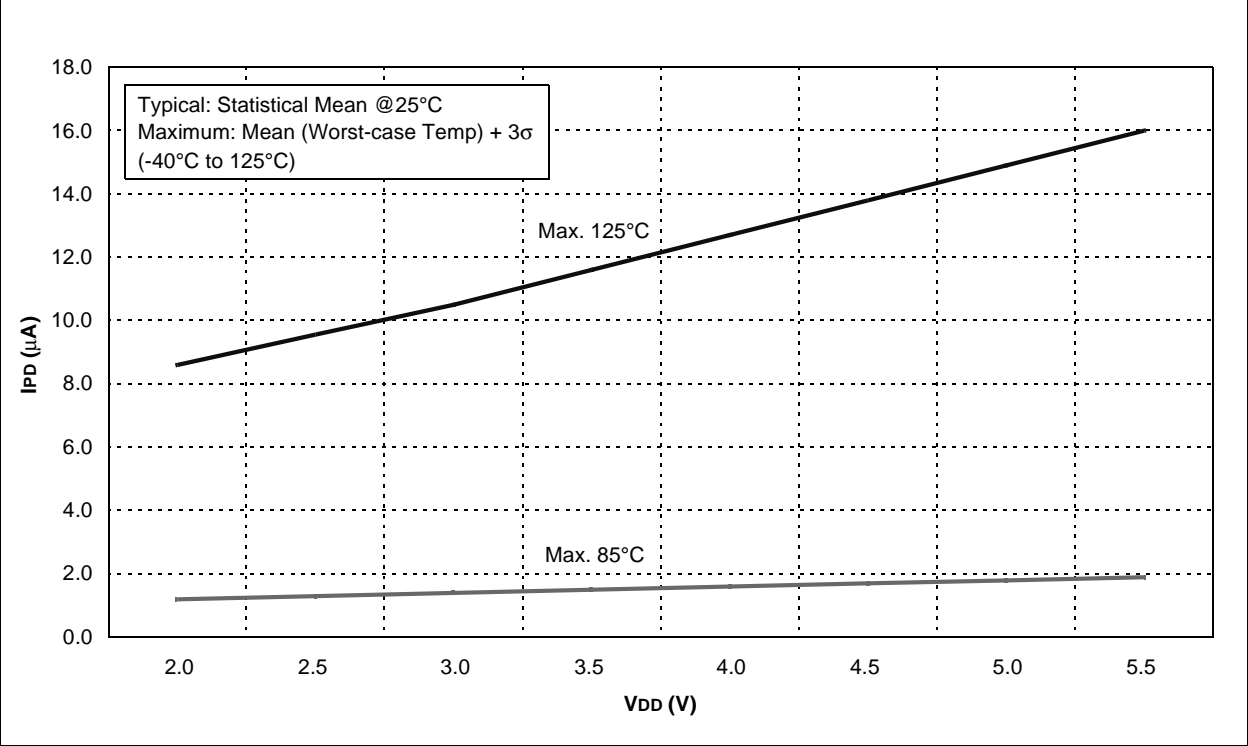


FIGURE 16-15: COMPARATOR  $I_{PD}$  vs.  $V_{DD}$  (BOTH COMPARATORS ENABLED)

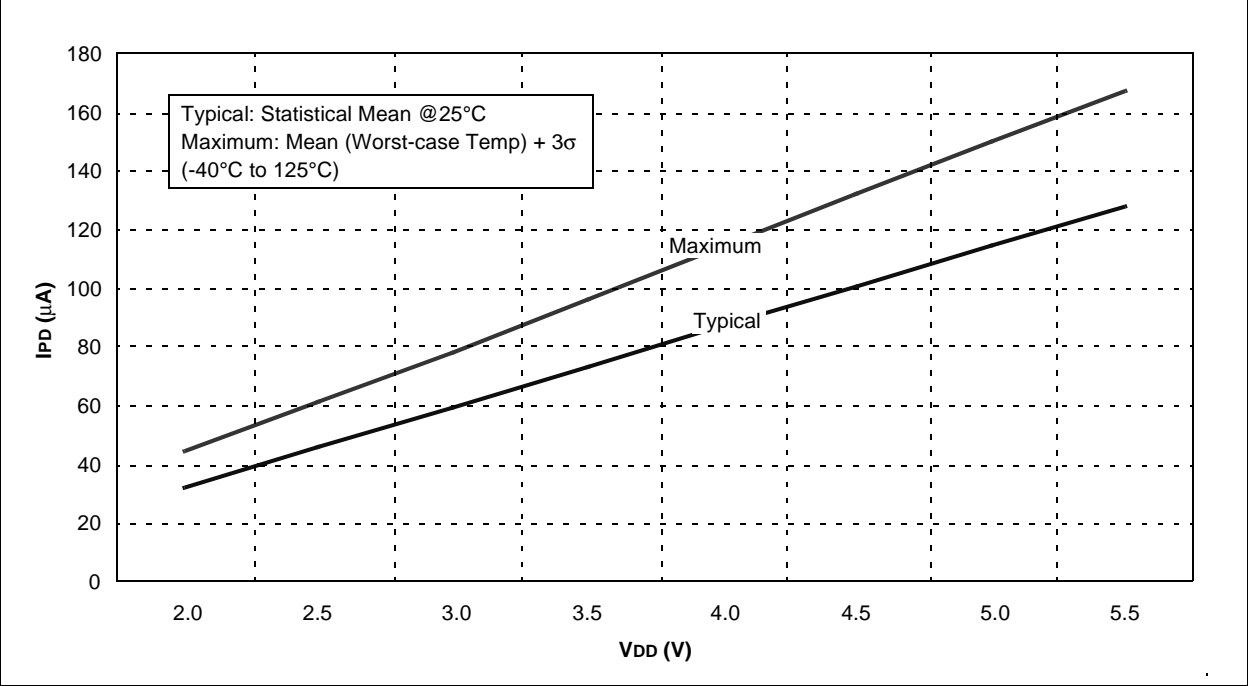


FIGURE 16-16: BOR IPD vs. VDD OVER TEMPERATURE

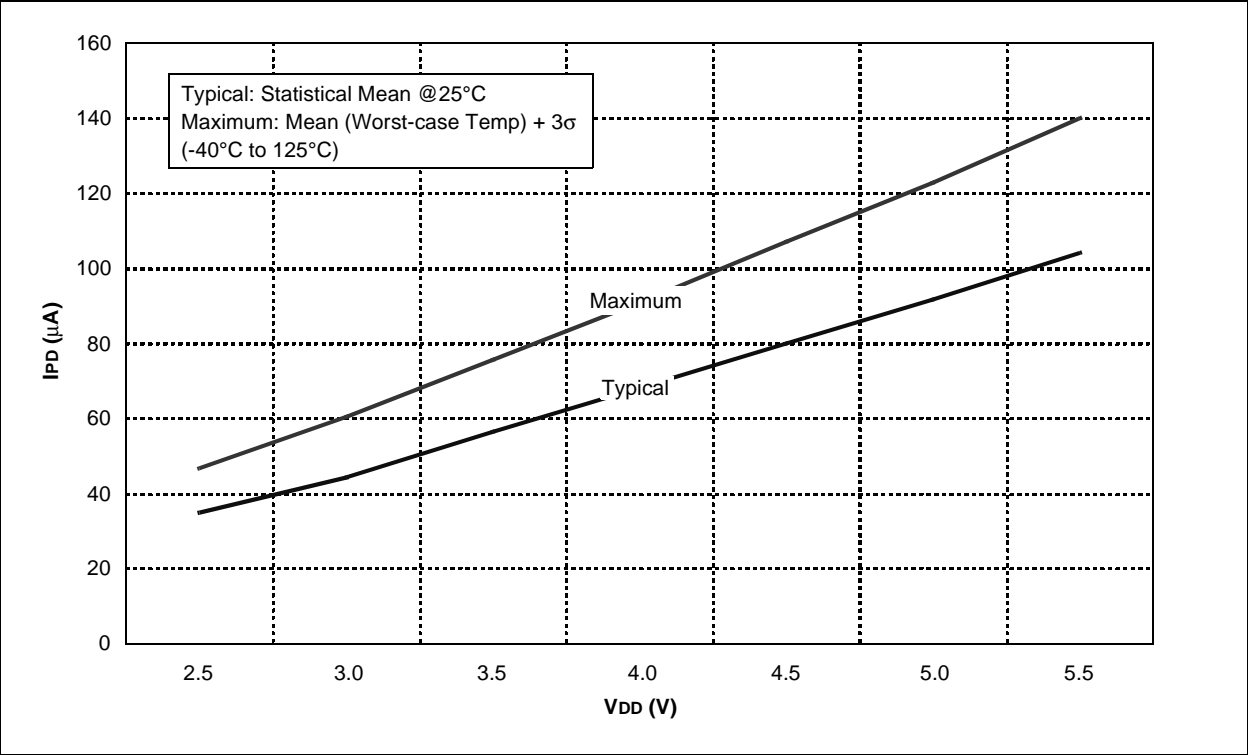
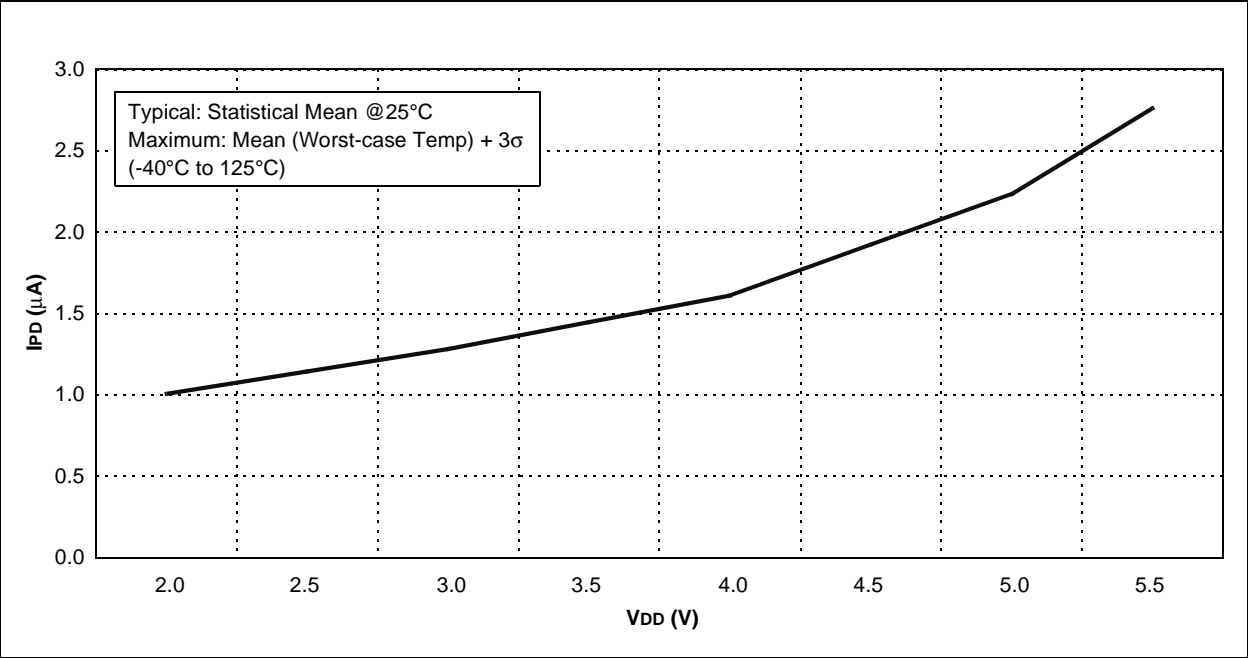


FIGURE 16-17: TYPICAL WDT IPD vs. VDD OVER TEMPERATURE



# PIC12F683

FIGURE 16-18: MAXIMUM WDT IPD vs. VDD OVER TEMPERATURE

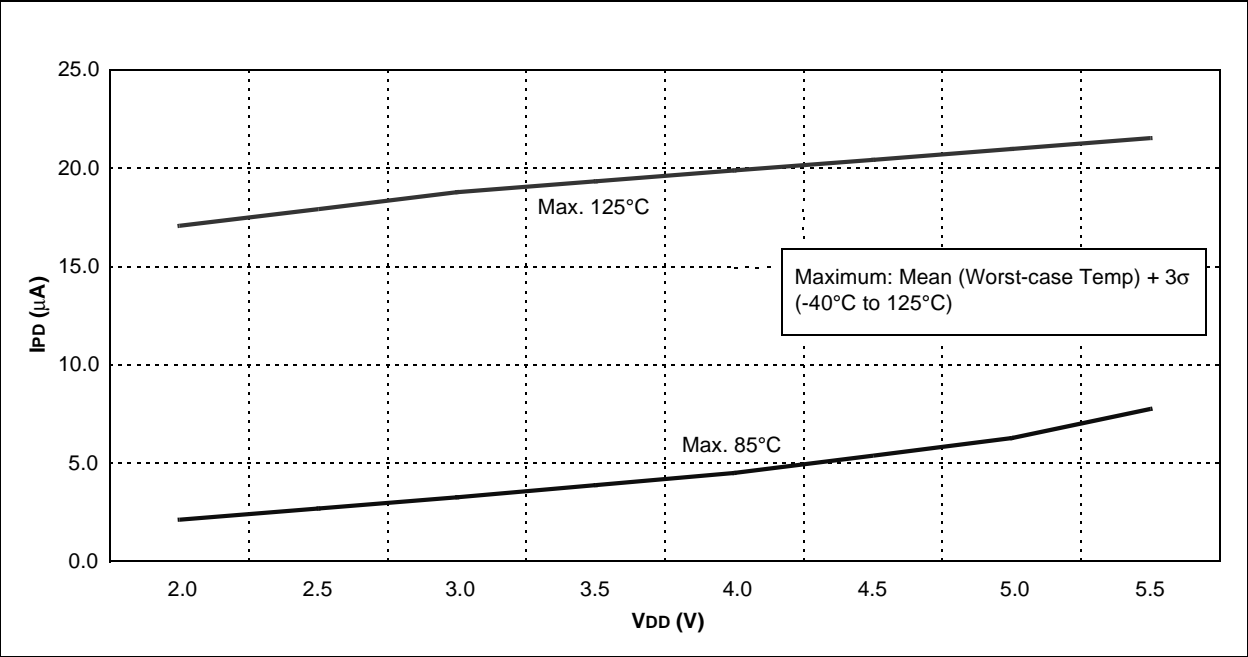
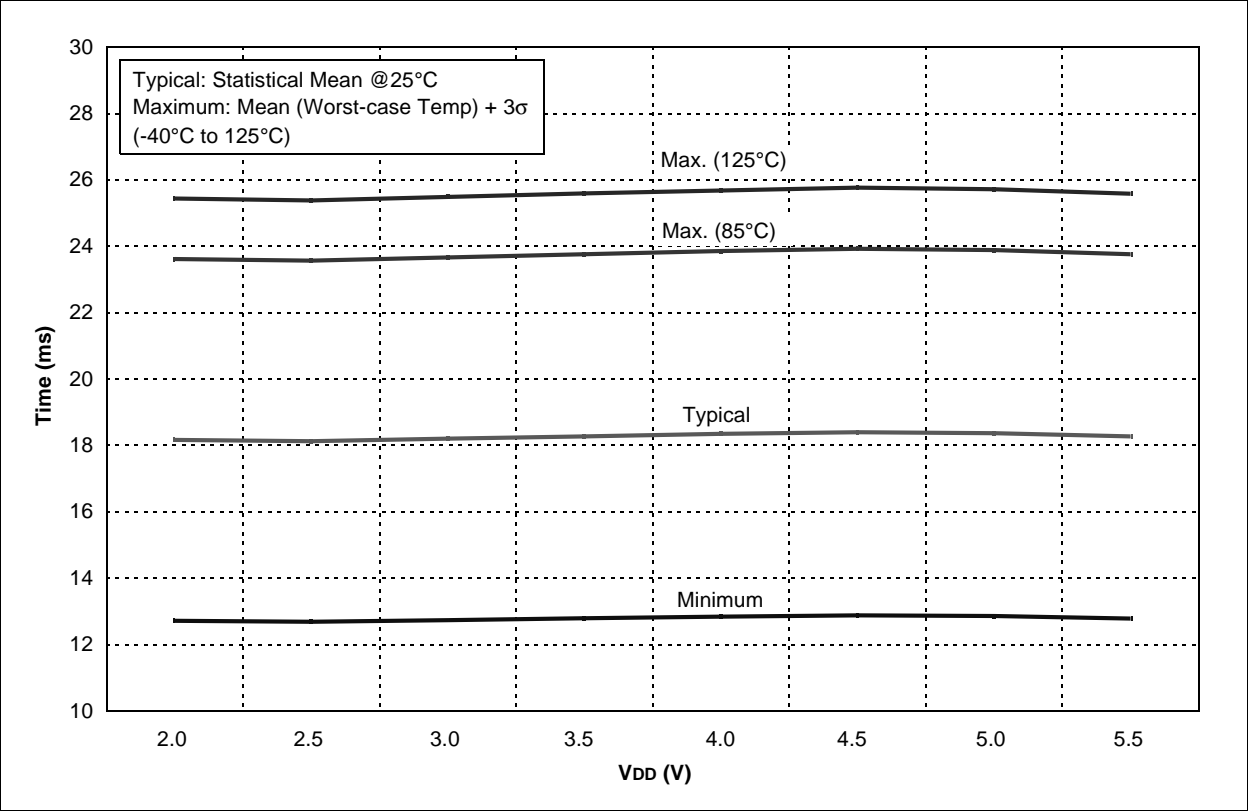
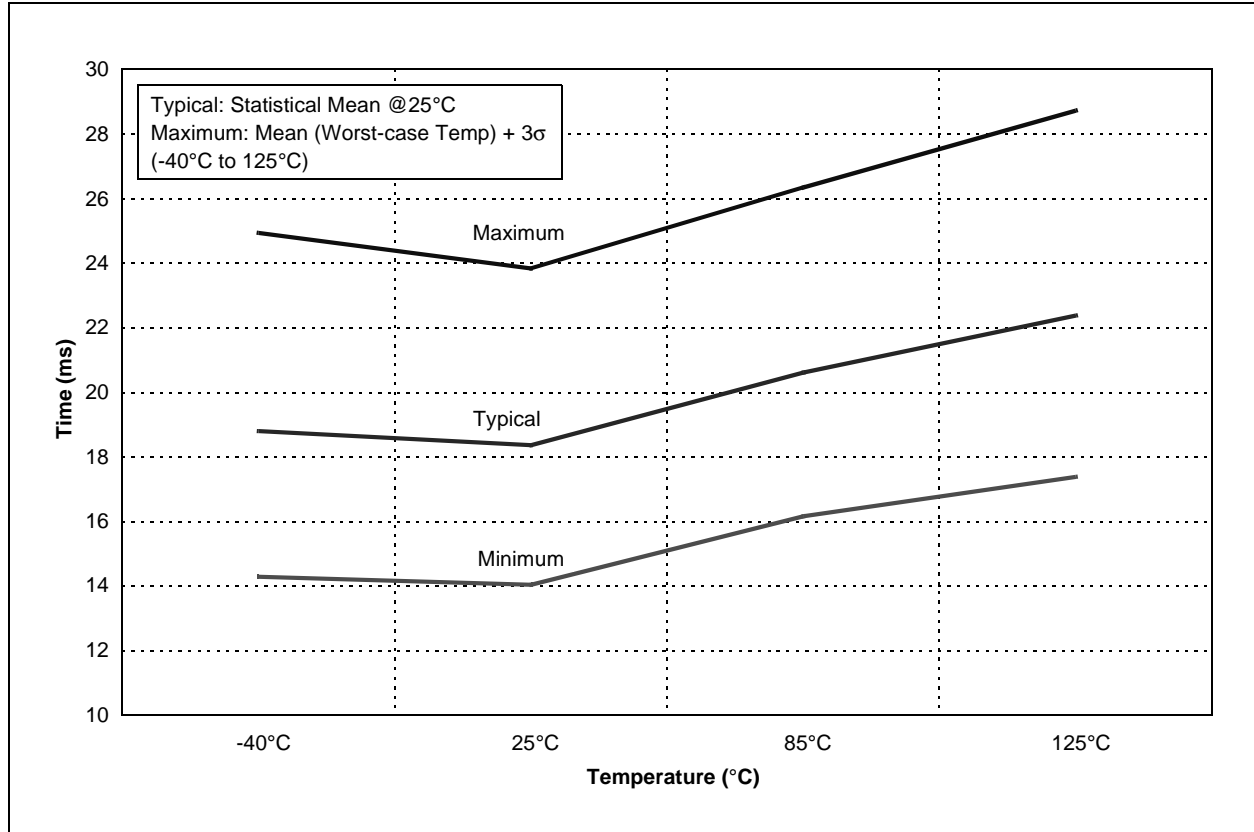


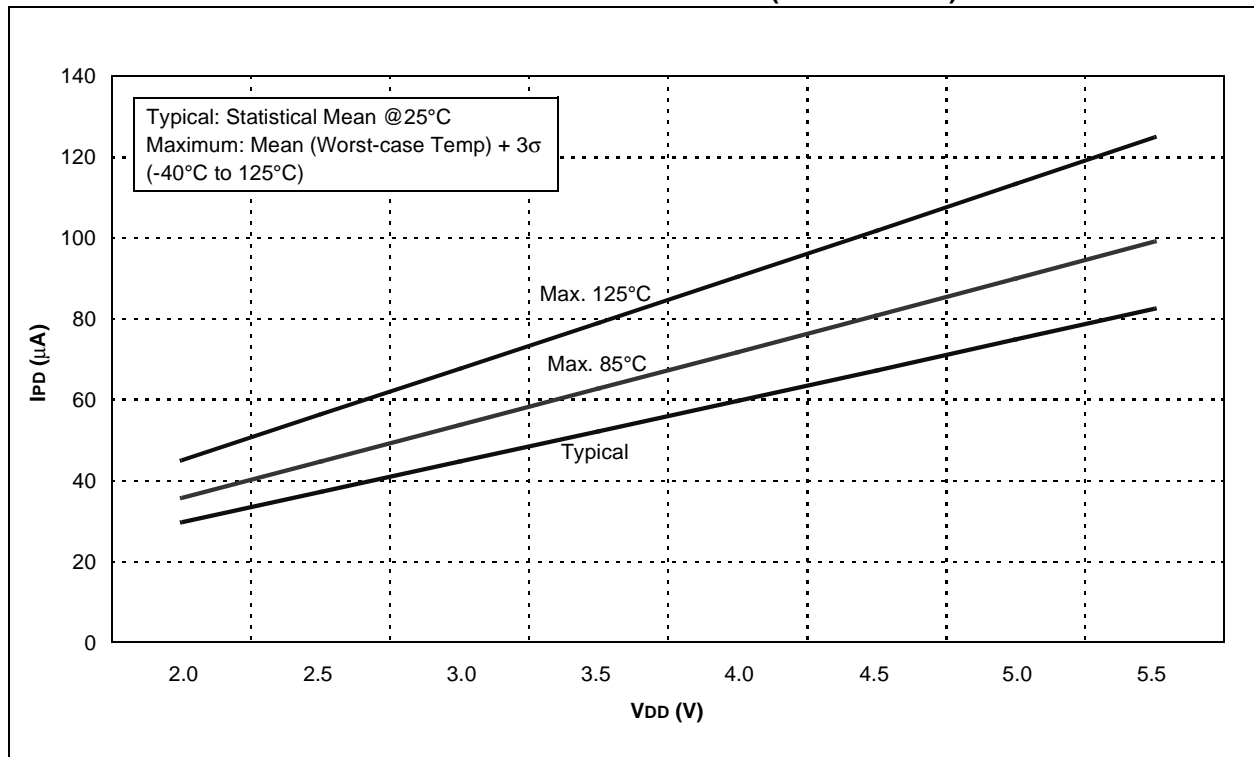
FIGURE 16-19: WDT PERIOD vs. VDD OVER TEMPERATURE



**FIGURE 16-20: WDT PERIOD vs. TEMPERATURE OVER V<sub>DD</sub> (5.0V)**



**FIGURE 16-21: CV<sub>REF</sub> IPD vs. V<sub>DD</sub> OVER TEMPERATURE (HIGH RANGE)**



# PIC12F683

FIGURE 16-22: CVREF IPD vs. VDD OVER TEMPERATURE (LOW RANGE)

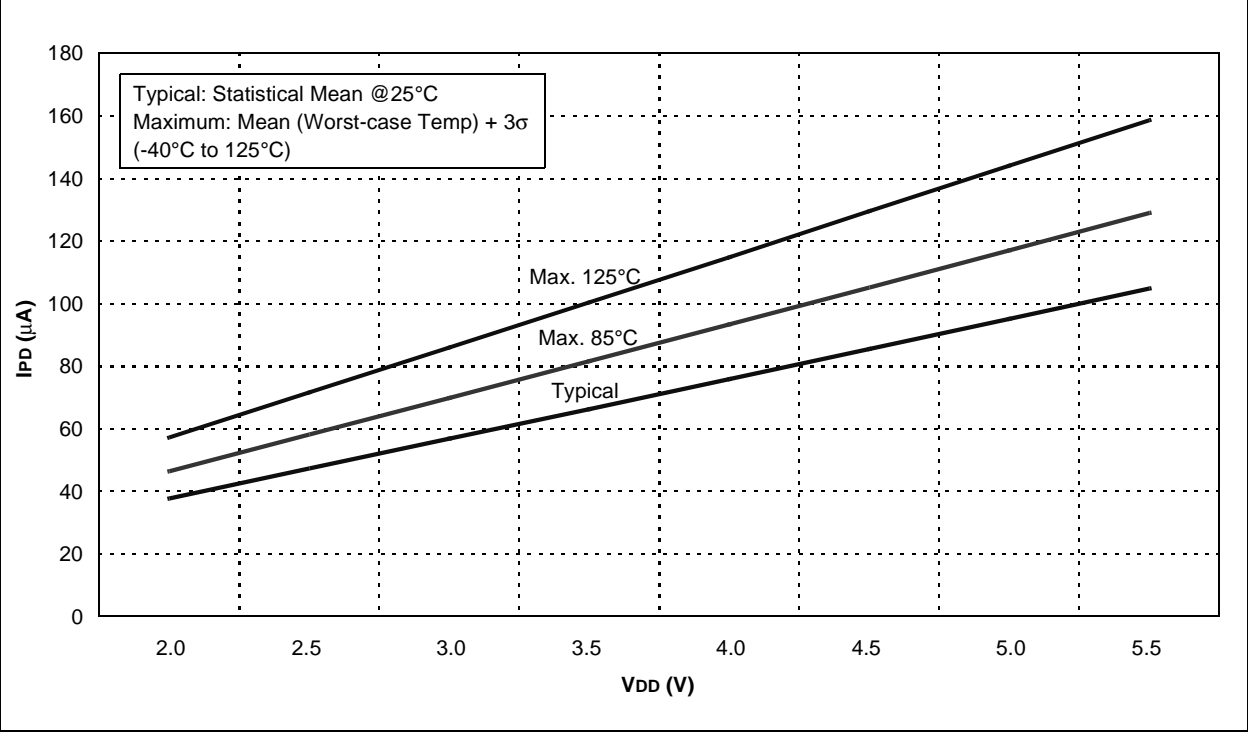


FIGURE 16-23: VOL vs. IOL OVER TEMPERATURE (VDD = 3.0V)

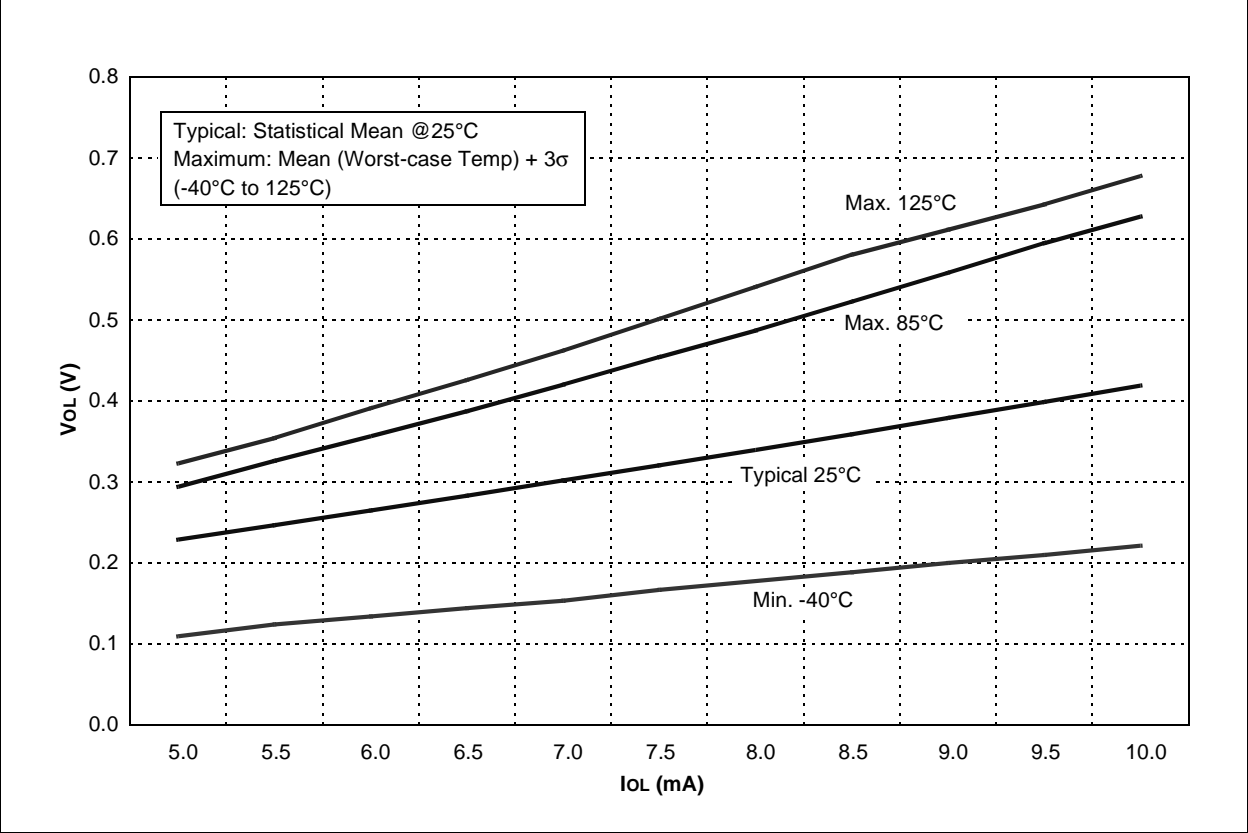




FIGURE 16-24: **VoL vs. IoL OVER TEMPERATURE (VDD = 5.0V)**

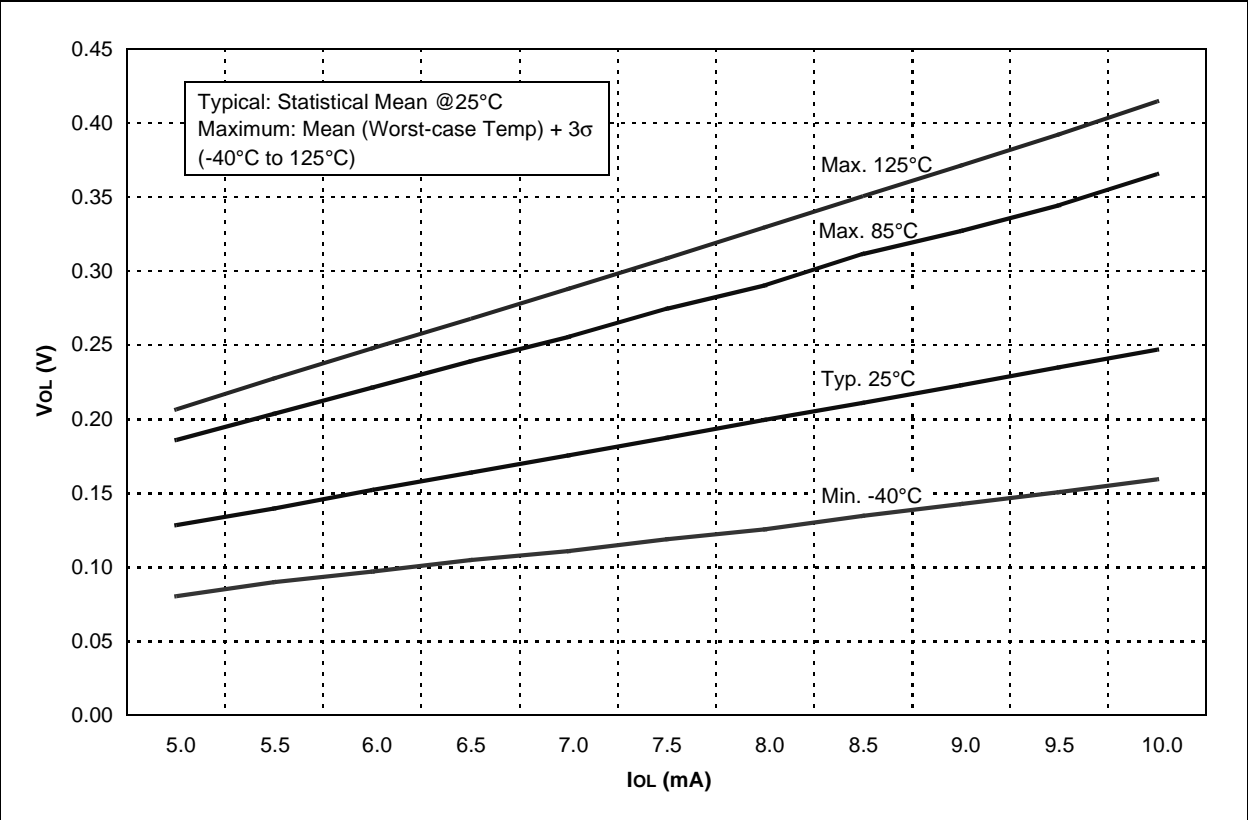


FIGURE 16-25: **VoH vs. IoH OVER TEMPERATURE (VDD = 3.0V)**

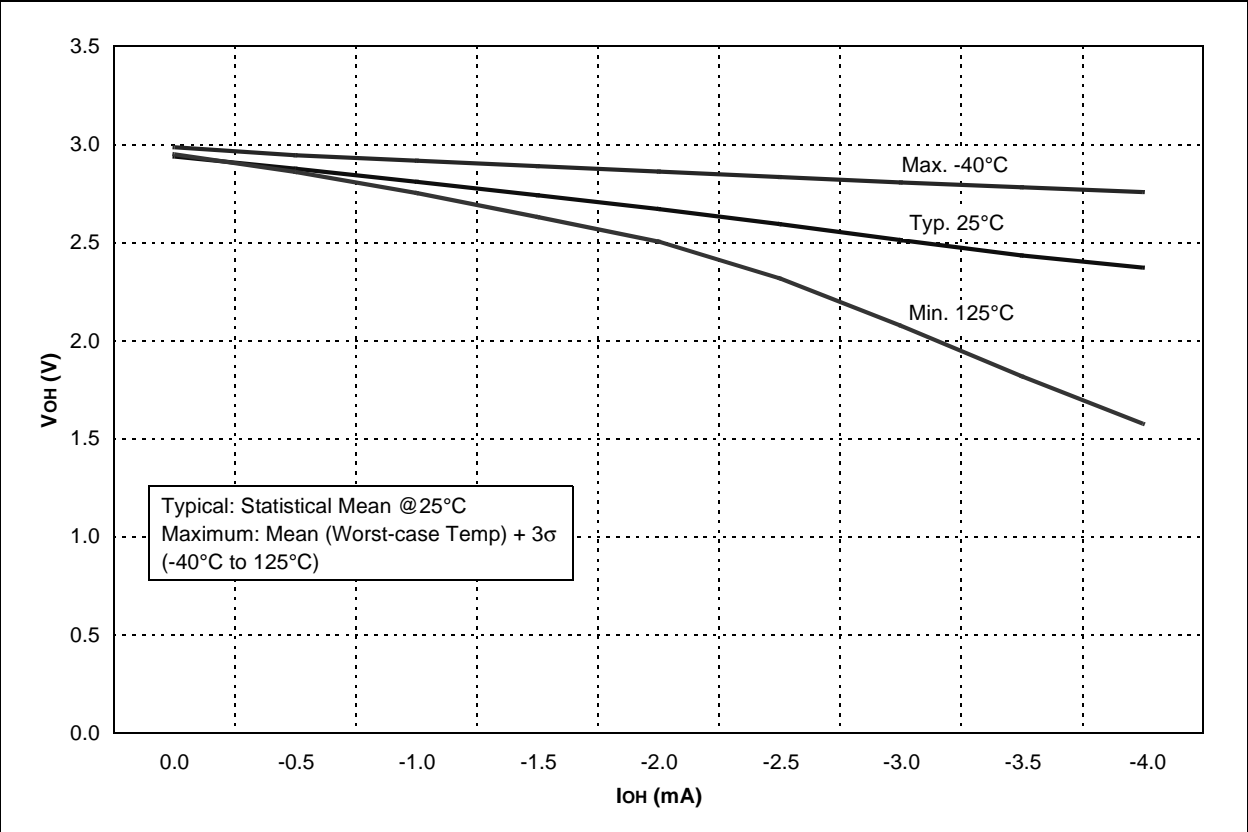


FIGURE 16-26:  $V_{OH}$  vs.  $I_{OH}$  OVER TEMPERATURE ( $V_{DD} = 5.0V$ )

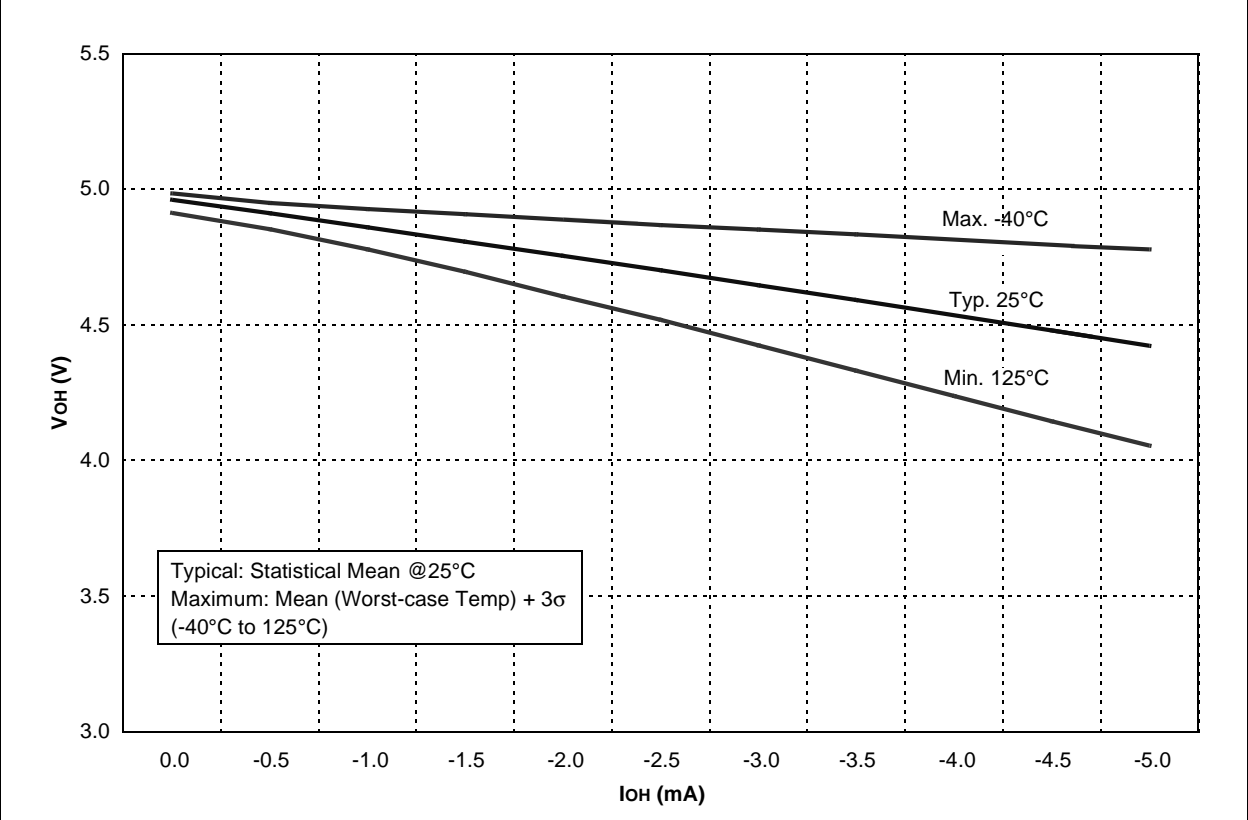


FIGURE 16-27: TTL INPUT THRESHOLD  $V_{IN}$  vs.  $V_{DD}$  OVER TEMPERATURE

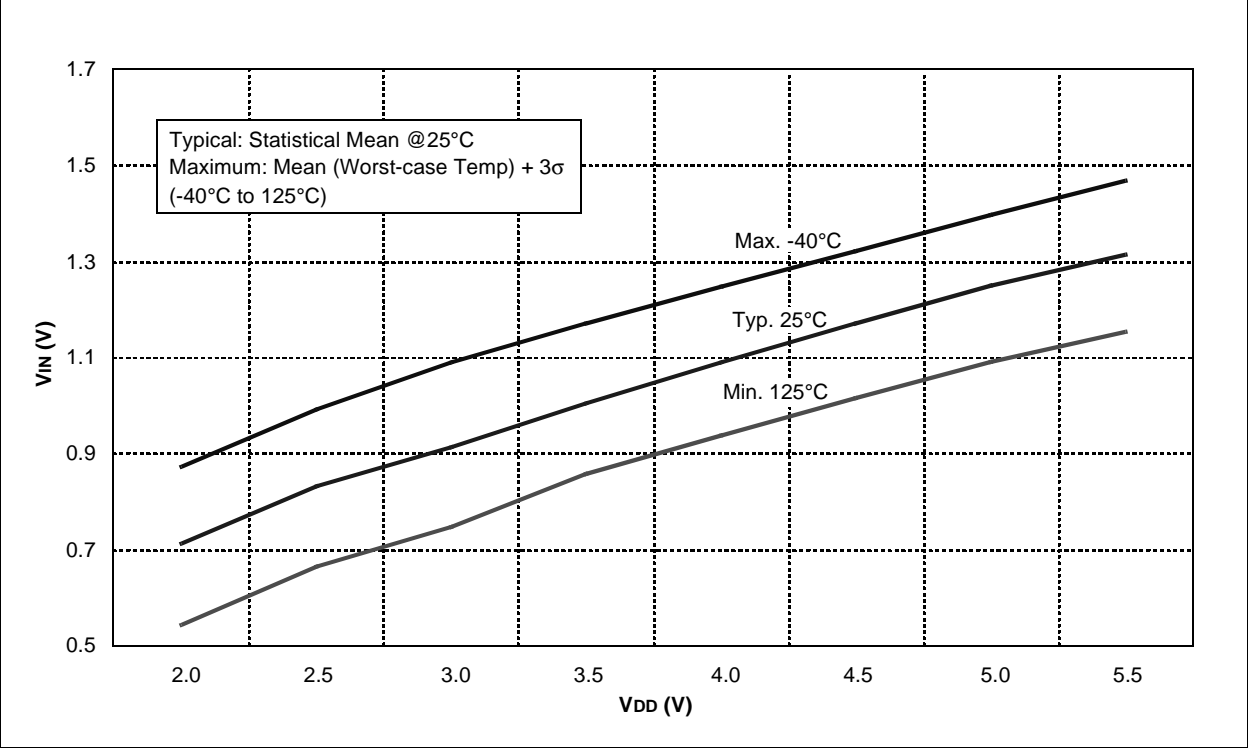


FIGURE 16-28: SCHMITT TRIGGER INPUT THRESHOLD VIN vs. VDD OVER TEMPERATURE

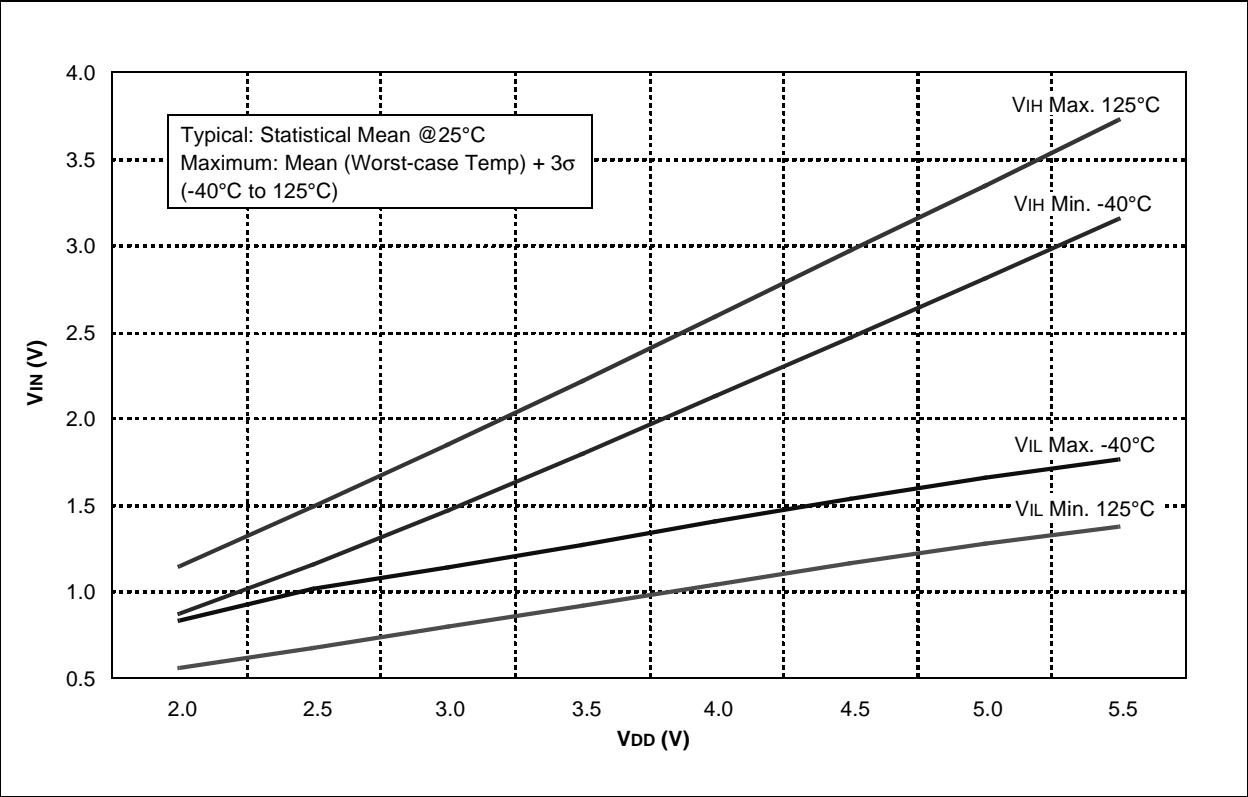
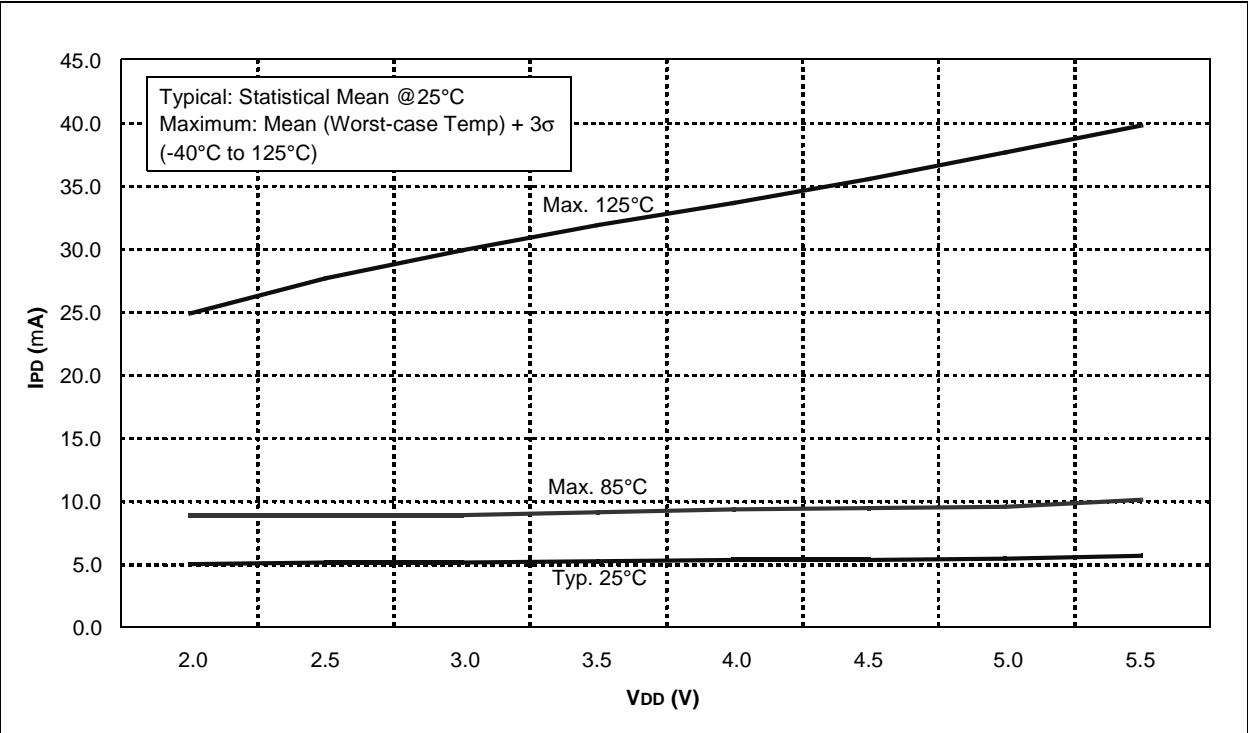


FIGURE 16-29: T1OSC IPD vs. VDD OVER TEMPERATURE (32 kHz)



# PIC12F683

FIGURE 16-30: COMPARATOR RESPONSE TIME (RISING EDGE)

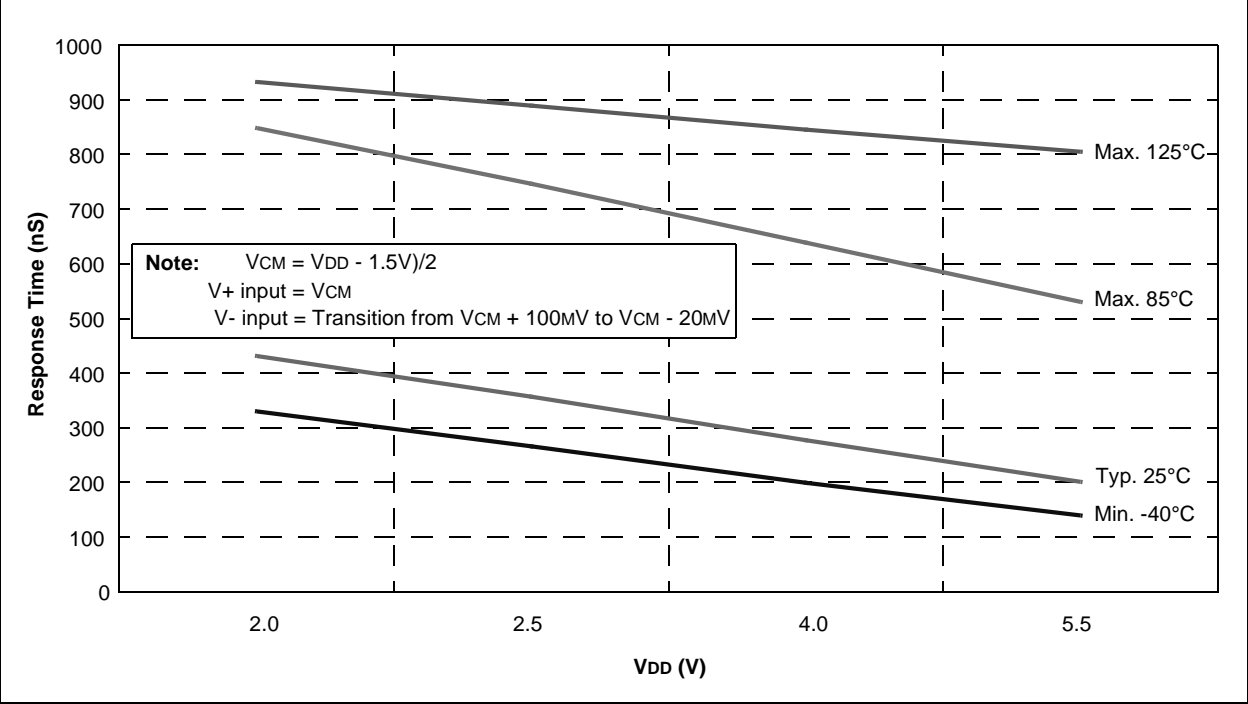
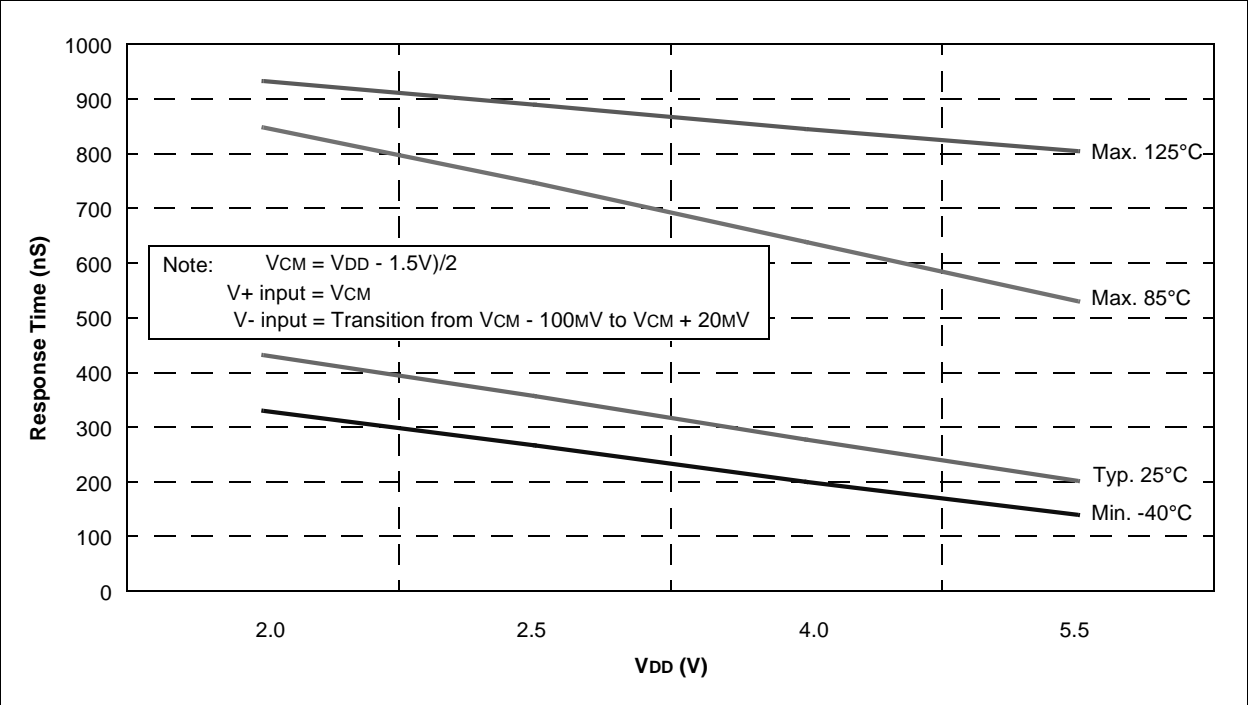
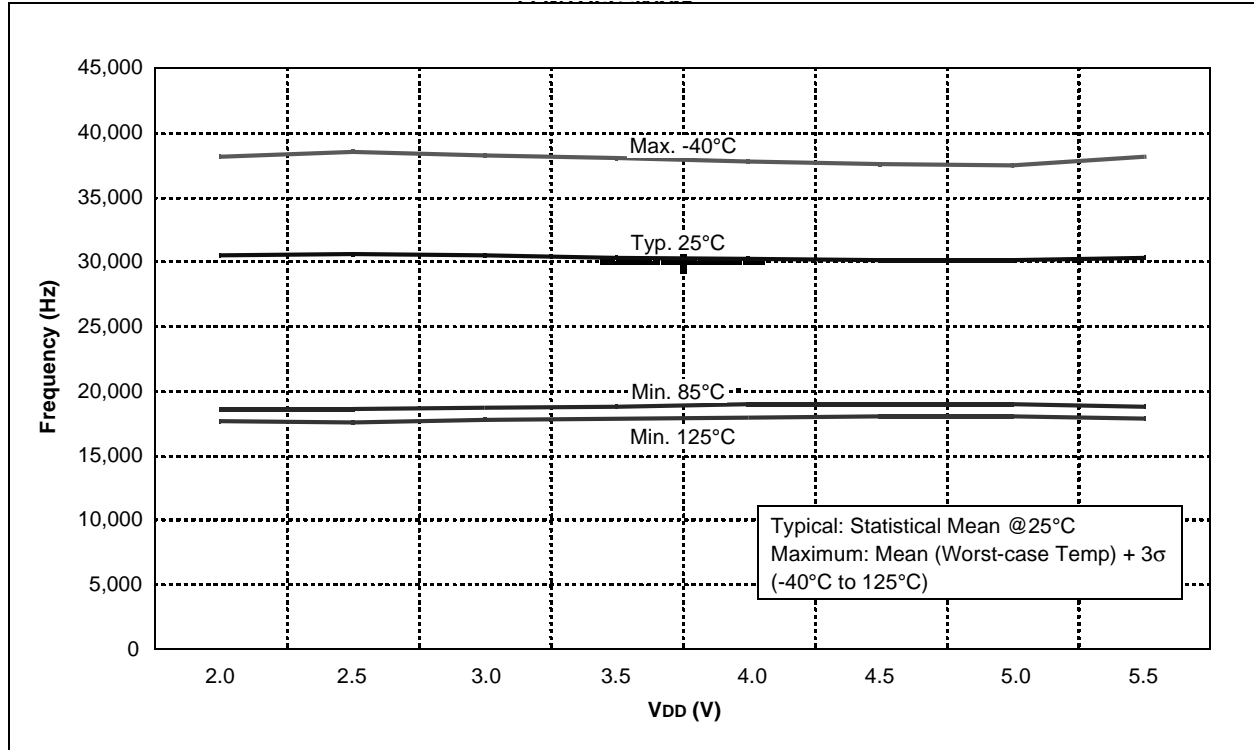


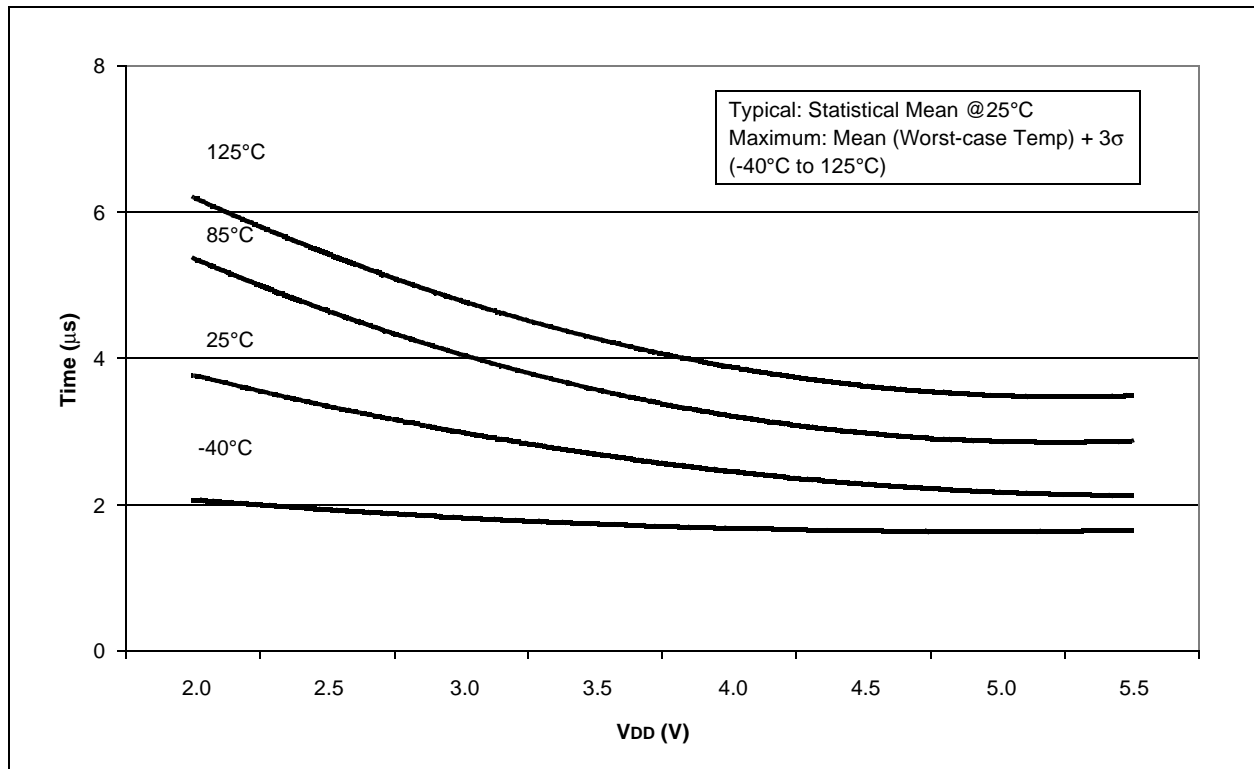
FIGURE 16-31: COMPARATOR RESPONSE TIME (FALLING EDGE)



**FIGURE 16-32: LFINTOSC FREQUENCY vs. V<sub>DD</sub> OVER TEMPERATURE (31 kHz)**



**FIGURE 16-33: ADC CLOCK PERIOD vs. V<sub>DD</sub> OVER TEMPERATURE**



# PIC12F683

FIGURE 16-34: TYPICAL HFINTOSC START-UP TIMES vs. VDD OVER TEMPERATURE

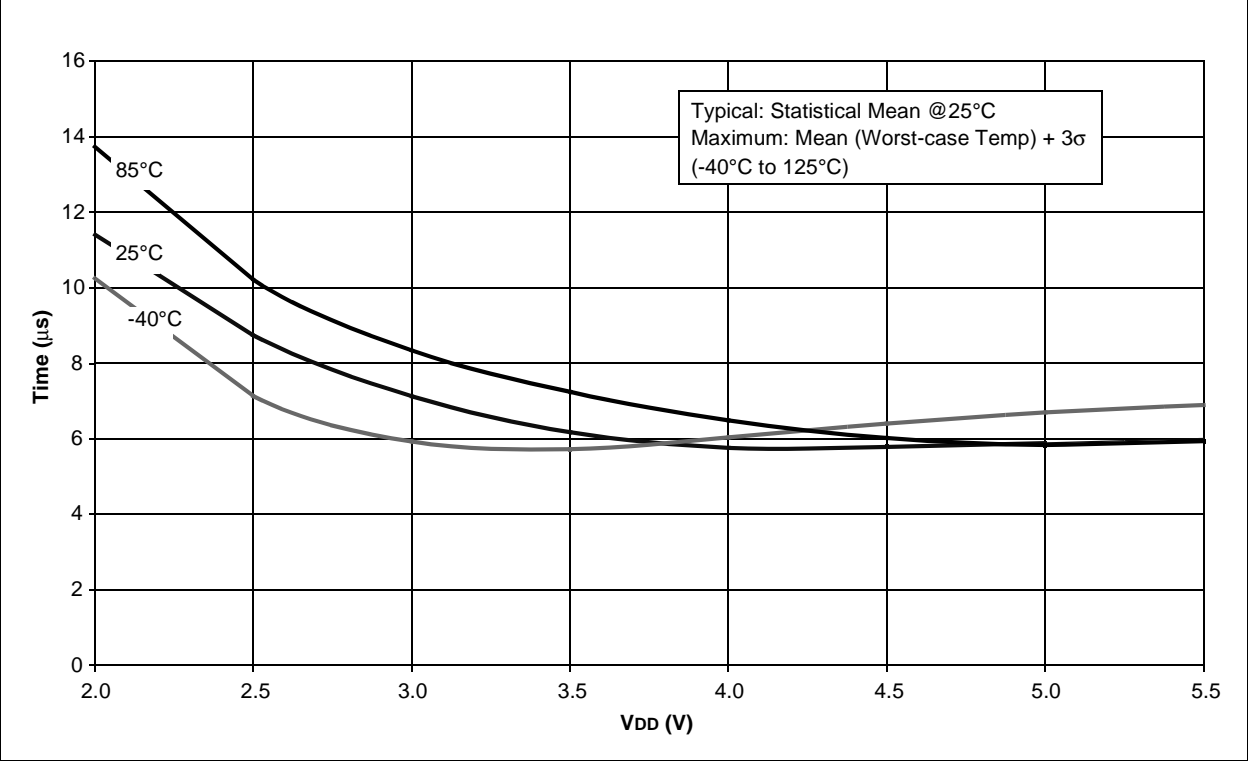
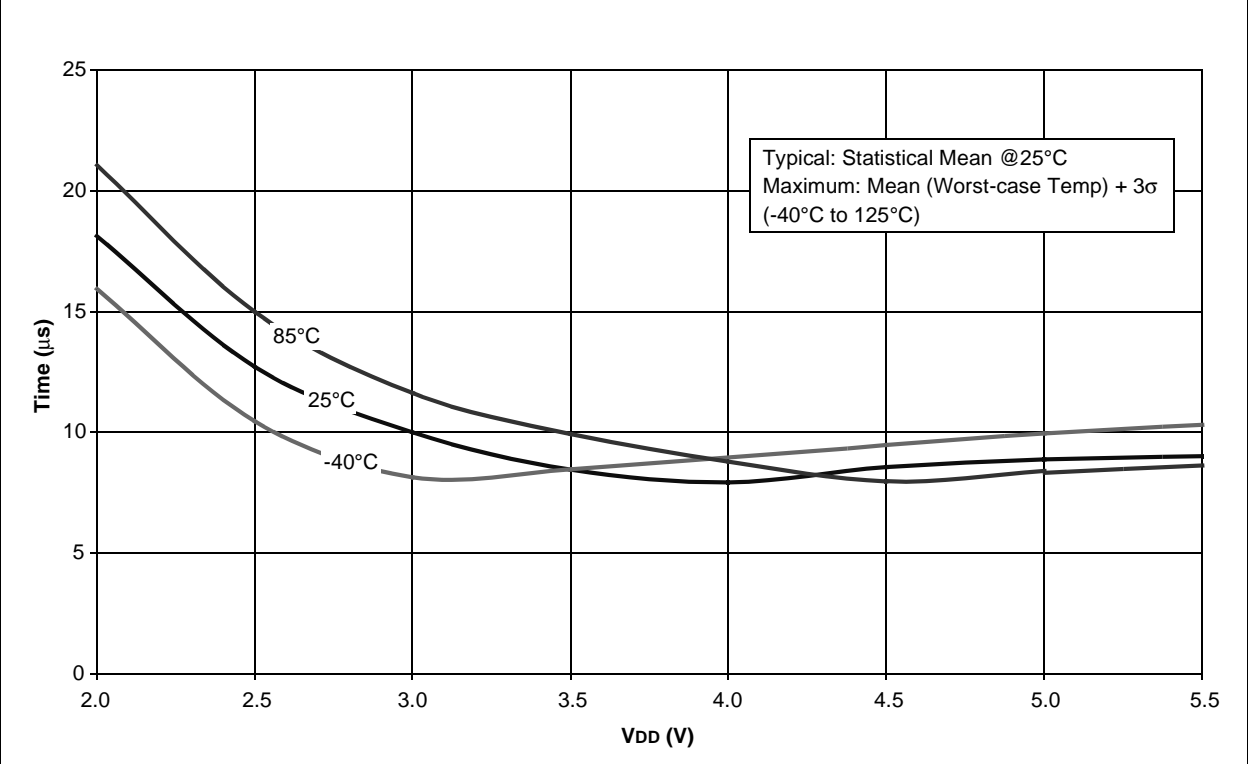
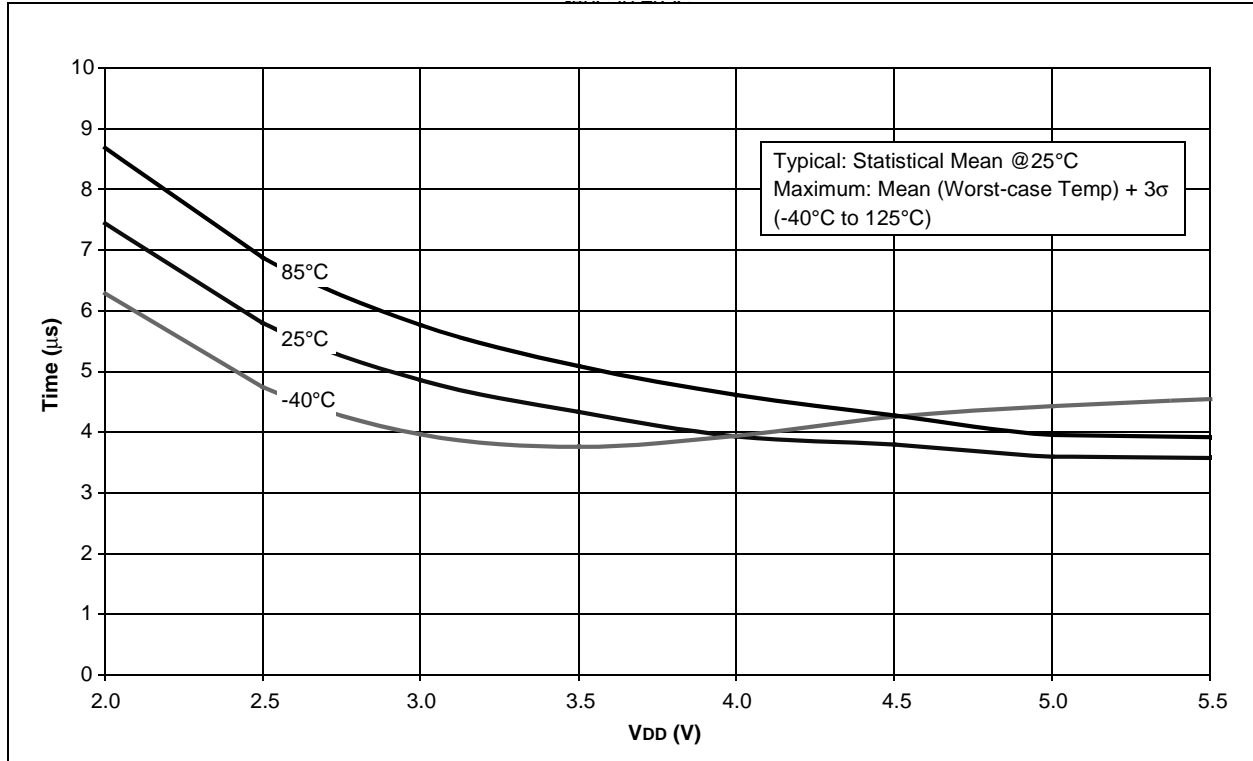


FIGURE 16-35: MAXIMUM HFINTOSC START-UP TIMES vs. VDD OVER TEMPERATURE



**FIGURE 16-36: MINIMUM HFINTOSC START-UP TIMES vs. V<sub>DD</sub> OVER TEMPERATURE**



**FIGURE 16-37: TYPICAL HFINTOSC FREQUENCY CHANGE vs. V<sub>DD</sub> (25°C)**

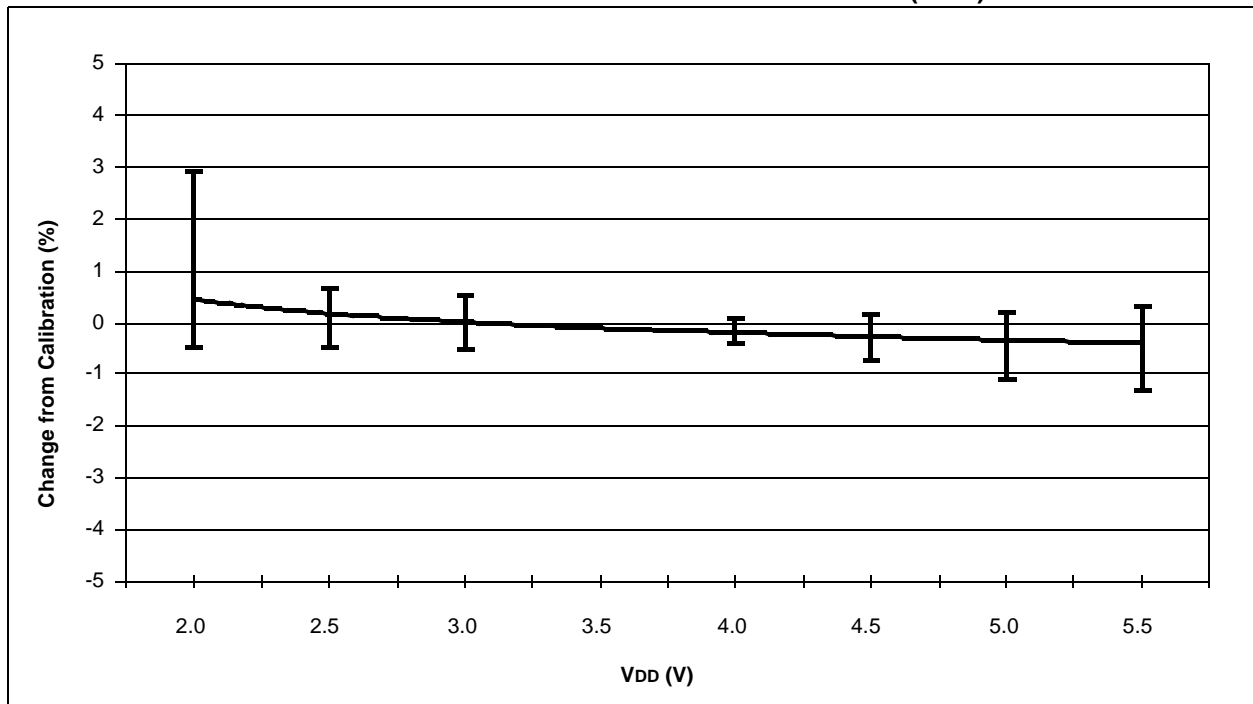


FIGURE 16-38: TYPICAL HFINTOSC FREQUENCY CHANGE OVER DEVICE VDD (85°C)

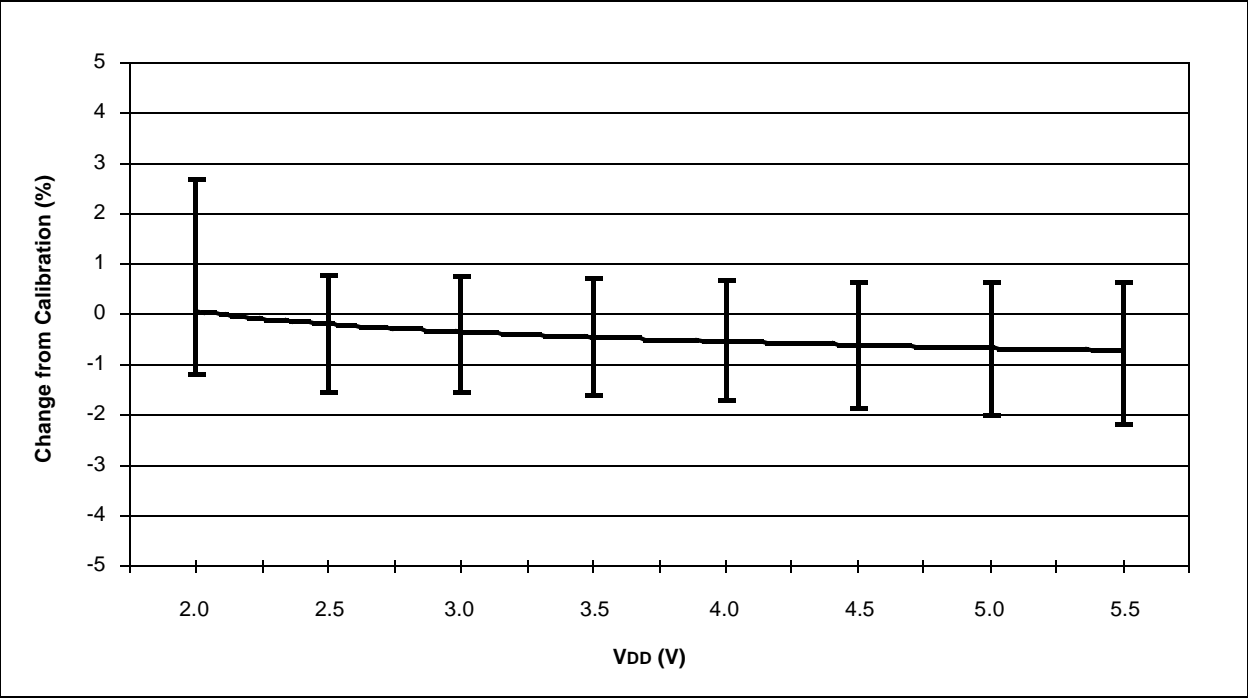
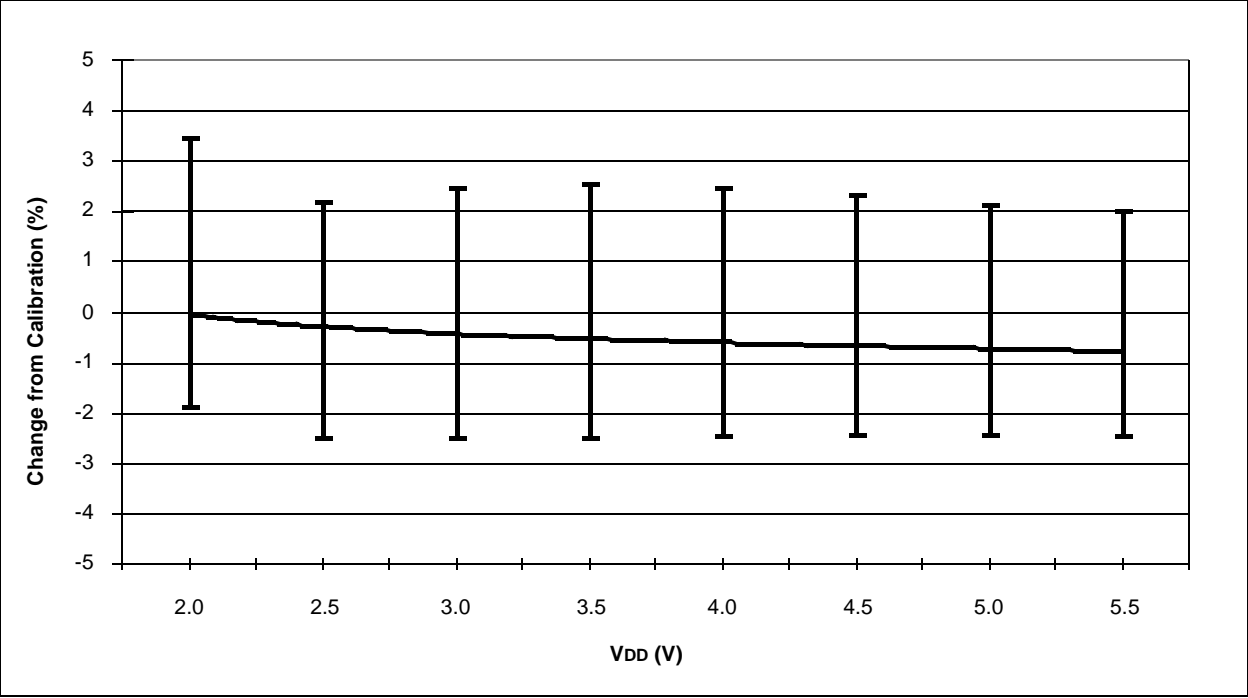
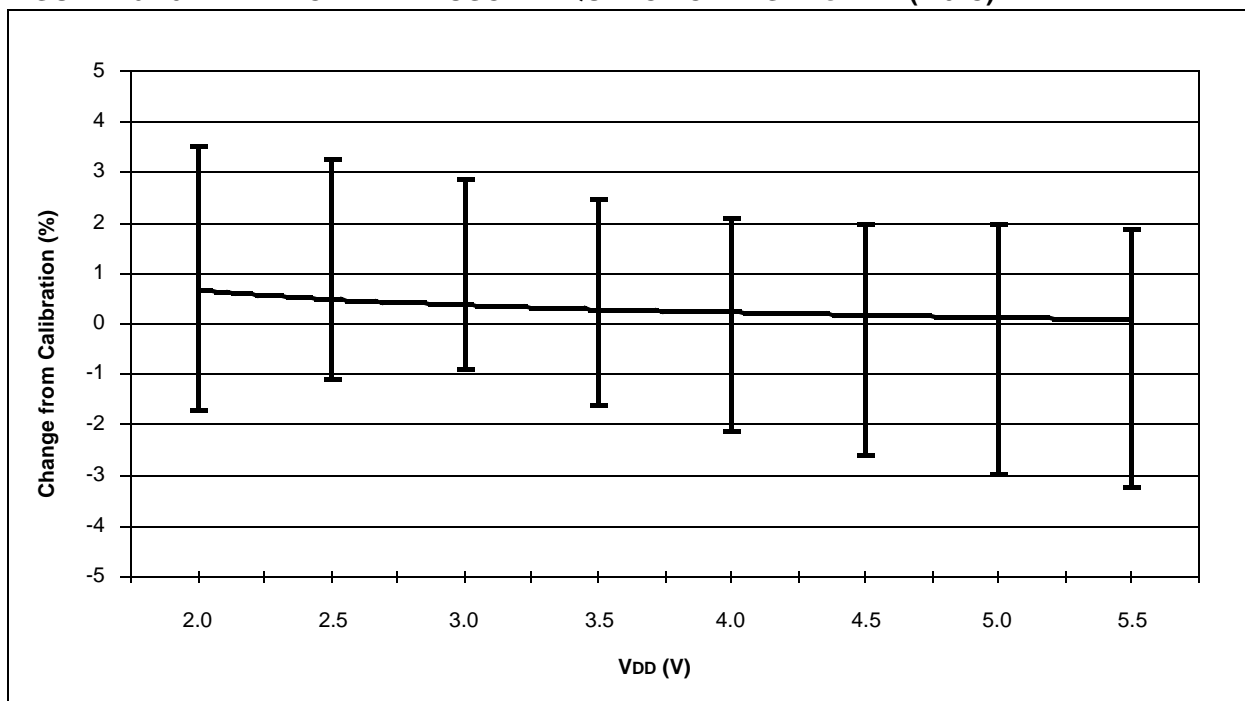


FIGURE 16-39: TYPICAL HFINTOSC FREQUENCY CHANGE vs. VDD (125°C)





**FIGURE 16-40: TYPICAL HFINTOSC FREQUENCY CHANGE vs.  $V_{DD}$  ( $-40^{\circ}\text{C}$ )**



# PIC12F683

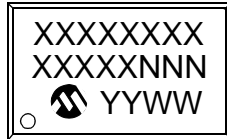
---

NOTES:

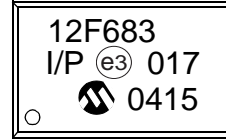
## 17.0 PACKAGING INFORMATION

### 17.1 Package Marking Information

8-Lead PDIP



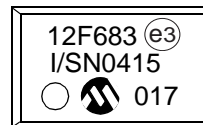
Example



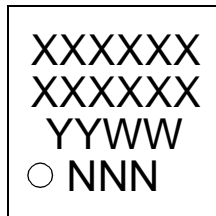
8-Lead SOIC (3.90 mm)



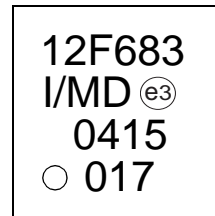
Example



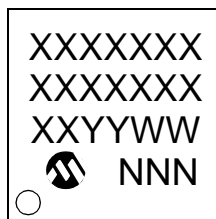
8-Lead DFN (4x4x0.9 mm)



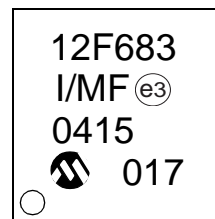
Example



8-Lead DFN-S (6x5 mm)



Example



**Legend:** XX...X Customer-specific information  
Y Year code (last digit of calendar year)  
YY Year code (last 2 digits of calendar year)  
WW Week code (week of January 1 is week '01')  
NNN Alphanumeric traceability code  
(e3) Pb-free JEDEC designator for Matte Tin (Sn)  
\* This package is Pb-free. The Pb-free JEDEC designator ((e3)) can be found on the outer packaging for this package.

**Note:** In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information.

\* Standard PIC® device marking consists of Microchip part number, year code, week code and traceability code. For PIC device marking beyond this, certain price adders apply. Please check with your Microchip Sales Office. For QTP devices, any special marking adders are included in QTP price.

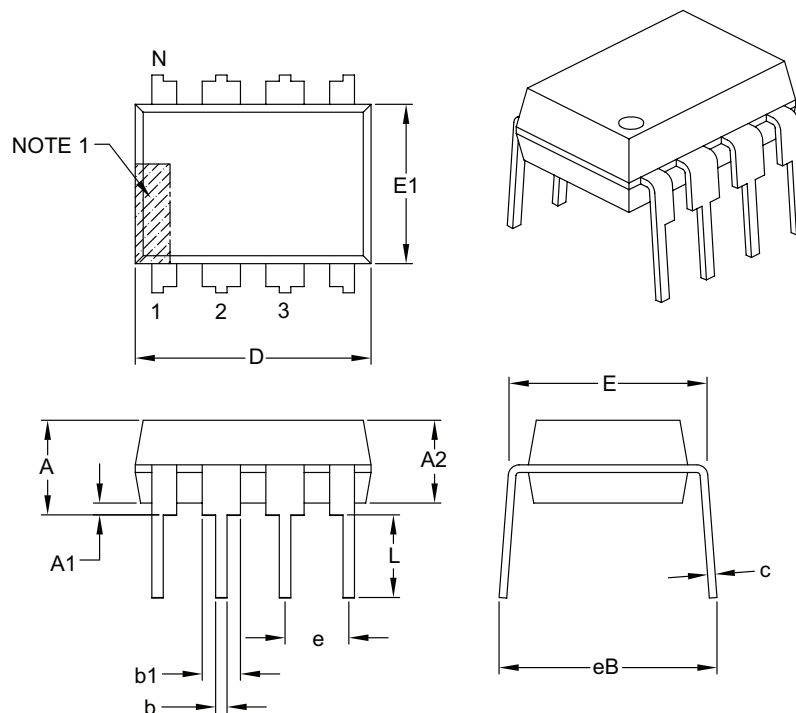
# PIC12F683

## 17.2 Package Details

The following sections give the technical details of the packages.

### 8-Lead Plastic Dual In-Line (P or PA) – 300 mil Body [PDIP]

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



Units		INCHES		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N	8		
Pitch	e	.100 BSC		
Top to Seating Plane	A	–	–	.210
Molded Package Thickness	A2	.115	.130	.195
Base to Seating Plane	A1	.015	–	–
Shoulder to Shoulder Width	E	.290	.310	.325
Molded Package Width	E1	.240	.250	.280
Overall Length	D	.348	.365	.400
Tip to Seating Plane	L	.115	.130	.150
Lead Thickness	c	.008	.010	.015
Upper Lead Width	b1	.040	.060	.070
Lower Lead Width	b	.014	.018	.022
Overall Row Spacing §	eB	–	–	.430

**Notes:**

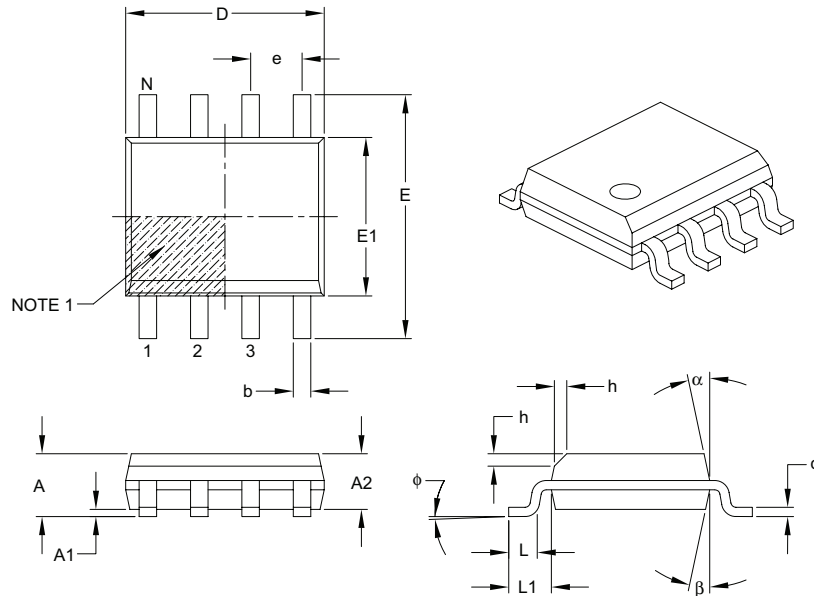
- Pin 1 visual index feature may vary, but must be located with the hatched area.
- § Significant Characteristic.
- Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.
- Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing C04-018B

## 8-Lead Plastic Small Outline (SN or OA) – Narrow, 3.90 mm Body [SOIC]

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N	8		
Pitch	e	1.27 BSC		
Overall Height	A	–	–	1.75
Molded Package Thickness	A2	1.25	–	–
Standoff §	A1	0.10	–	0.25
Overall Width	E	6.00 BSC		
Molded Package Width	E1	3.90 BSC		
Overall Length	D	4.90 BSC		
Chamfer (optional)	h	0.25	–	0.50
Foot Length	L	0.40	–	1.27
Footprint	L1	1.04 REF		
Foot Angle	φ	0°	–	8°
Lead Thickness	c	0.17	–	0.25
Lead Width	b	0.31	–	0.51
Mold Draft Angle Top	α	5°	–	15°
Mold Draft Angle Bottom	β	5°	–	15°

### Notes:

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- § Significant Characteristic.
- Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.15 mm per side.
- Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

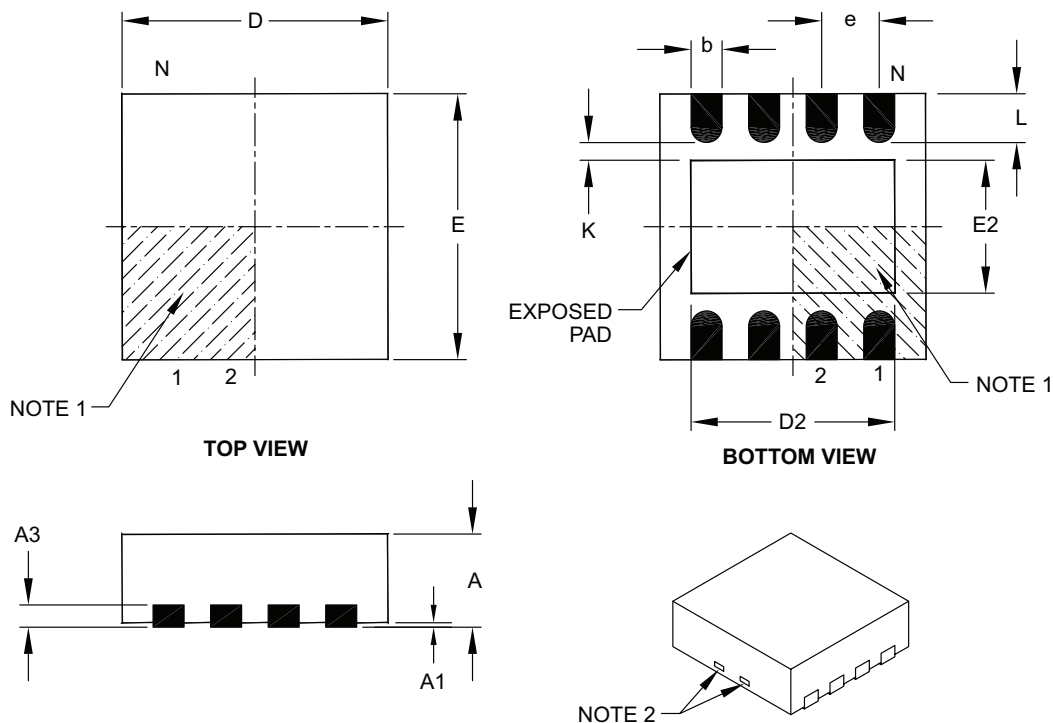
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-057B

# PIC12F683

## 8-Lead Plastic Dual Flat, No Lead Package (MD) – 4x4x0.9 mm Body [DFN]

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packageing>



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N	8		
Pitch	e	0.80 BSC		
Overall Height	A	0.80	0.90	1.00
Standoff	A1	0.00	0.02	0.05
Contact Thickness	A3	0.20 REF		
Overall Length	D	4.00 BSC		
Exposed Pad Width	E2	0.00	2.20	2.80
Overall Width	E	4.00 BSC		
Exposed Pad Length	D2	0.00	3.00	3.60
Contact Width	b	0.25	0.30	0.35
Contact Length	L	0.30	0.55	0.65
Contact-to-Exposed Pad	K	0.20	—	—

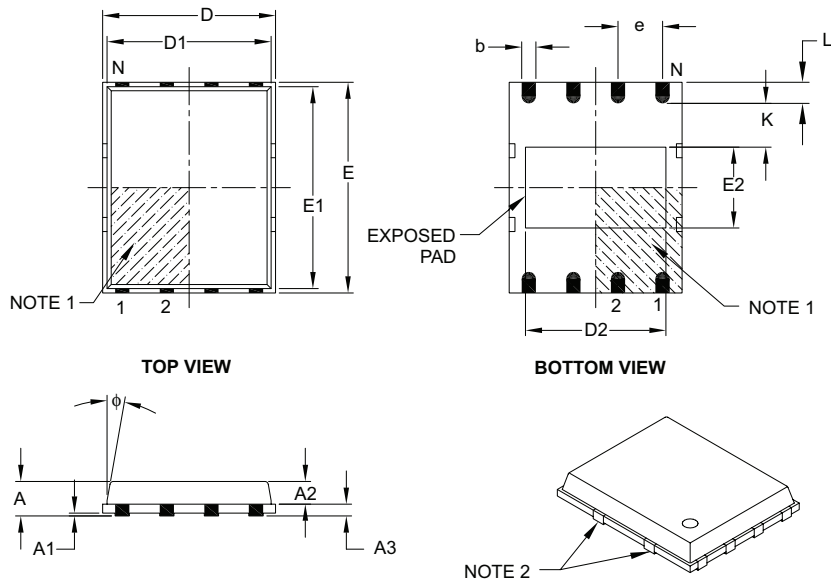
### Notes:

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- Package may have one or more exposed tie bars at ends.
- Package is saw singulated.
- Dimensioning and tolerancing per ASME Y14.5M.  
BSC: Basic Dimension. Theoretically exact value shown without tolerances.  
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-131C

## 8-Lead Plastic Dual Flat, No Lead Package (MF) – 6x5 mm Body [DFN-S]

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N	8		
Pitch	e	1.27 BSC		
Overall Height	A	–	0.85	1.00
Molded Package Thickness	A2	–	0.65	0.80
Standoff	A1	0.00	0.01	0.05
Base Thickness	A3	0.20 REF		
Overall Length	D	4.92 BSC		
Molded Package Length	D1	4.67 BSC		
Exposed Pad Length	D2	3.85	4.00	4.15
Overall Width	E	5.99 BSC		
Molded Package Width	E1	5.74 BSC		
Exposed Pad Width	E2	2.16	2.31	2.46
Contact Width	b	0.35	0.40	0.47
Contact Length	L	0.50	0.60	0.75
Contact-to-Exposed Pad	K	0.20	–	–
Model Draft Angle Top	φ	–	–	12°

**Notes:**

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- Package may have one or more exposed tie bars at ends.
- Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-113B

# PIC12F683

---

NOTES:



## APPENDIX A: DATA SHEET REVISION HISTORY

### Revision A

This is a new data sheet.

### Revision B

Rewrites of the Oscillator and Special Features of the CPU sections. General corrections to Figures and formatting.

### Revision C

Revisions throughout document. Incorporated Golden Chapters.

### Revision D

Replaced Package Drawings; Revised Product ID Section (SN package to 3.90 mm); Replaced PICmicro with PIC; Replaced Dev Tool Section.

## APPENDIX B: MIGRATING FROM OTHER PIC® DEVICES

This discusses some of the issues in migrating from other PIC devices to the PIC12F683 device.

### B.1 PIC16F676 to PIC12F683

**TABLE B-1: FEATURE COMPARISON**

Feature	PIC16F676	PIC12F683
Max Operating Speed	20 MHz	20 MHz
Max Program Memory (Words)	1024	2048
SRAM (bytes)	64	128
A/D Resolution	10-bit	10-bit
Data EEPROM (Bytes)	128	256
Timers (8/16-bit)	1/1	2/1
Oscillator Modes	8	8
Brown-out Reset	Y	Y
Internal Pull-ups	RA0/1/2/4/5	GP0/1/2/4/5, MCLR
Interrupt-on-change	RA0/1/2/3/4/5	GP0/1/2/3/4/5
Comparator	1	1
ECCP	N	N
Ultra Low-Power Wake-Up	N	Y
Extended WDT	N	Y
Software Control Option of WDT/BOR	N	Y
INTOSC Frequencies	4 MHz	32 kHz-8 MHz
Clock Switching	N	Y

**Note:** This device has been designed to perform to the parameters of its data sheet. It has been tested to an electrical specification designed to determine its conformance with these parameters. Due to process differences in the manufacture of this device, this device may have different performance characteristics than its earlier version. These differences may cause this device to perform differently in your application than the earlier version of this device.

# PIC12F683

---

NOTES:

## INDEX

### A

A/D	
Specifications .....	133, 134
Absolute Maximum Ratings .....	115
AC Characteristics	
Industrial and Extended .....	125
Load Conditions .....	124
ADC .....	61
Acquisition Requirements .....	67
Associated registers .....	69
Block Diagram .....	61
Calculating Acquisition Time .....	67
Channel Selection .....	61
Configuration .....	61
Configuring Interrupt .....	64
Conversion Clock .....	62
Conversion Procedure .....	64
GPIO Configuration .....	61
Internal Sampling Switch (Rss) IMPEDANCE .....	67
Interrupts .....	63
Operation .....	63
Operation During Sleep .....	64
Reference Voltage (VREF) .....	62
Result Formatting .....	63
Source Impedance .....	67
Special Event Trigger .....	64
Starting an A/D Conversion .....	63
ADCON0 Register .....	65
ADRESH Register (ADFM = 0) .....	66
ADRESH Register (ADFM = 1) .....	66
ADRESL Register (ADFM = 0) .....	66
ADRESL Register (ADFM = 1) .....	66
Analog Input Connection Considerations .....	52
Analog-to-Digital Converter. <i>See</i> ADC	
ANSEL Register .....	33
Assembler	
MPASM Assembler .....	112

### B

Block Diagrams	
(CCP) Capture Mode Operation .....	76
ADC .....	61
ADC Transfer Function .....	68
Analog Input Model .....	52, 68
CCP PWM .....	78
Clock Source .....	19
Comparator .....	51
Compare .....	77
Crystal Operation .....	22
External RC Mode .....	23
Fail-Safe Clock Monitor (FSCM) .....	29
GP1 Pin .....	37
GP2 Pin .....	37
GP3 Pin .....	38
GP4 Pin .....	38
GP5 Pin .....	39
In-Circuit Serial Programming Connections .....	100
Interrupt Logic .....	93
MCLR Circuit .....	86
On-Chip Reset Circuit .....	85
PIC12F683 .....	5
Resonator Operation .....	22
Timer1 .....	44

Timer2 .....	49
TMR0/WDT Prescaler .....	41
Watchdog Timer (WDT) .....	96
Brown-out Reset (BOR) .....	87
Associated .....	88
Calibration .....	87
Specifications .....	129
Timing and Characteristics .....	128

### C

C Compilers	
MPLAB C18 .....	112
MPLAB C30 .....	112
Calibration Bits .....	85
Capture Module. <i>See</i> Capture/Compare/PWM (CCP)	
Capture/Compare/PWM (CCP) .....	75
Associated registers w/ Capture, Compare and Timer1 .....	81
Associated registers w/ PWM and Timer2 .....	81
Capture Mode .....	76
CCPx Pin Configuration .....	76
Compare Mode .....	77
CCPx Pin Configuration .....	77
Software Interrupt Mode .....	76, 77
Special Event Trigger .....	77
Timer1 Mode Selection .....	76, 77
Prescaler .....	76
PWM Mode .....	78
Duty Cycle .....	79
Effects of Reset .....	80
Example PWM Frequencies and Resolutions, 20 MHz .....	79
Example PWM Frequencies and Resolutions, 8 MHz .....	79
Operation in Sleep Mode .....	80
Setup for Operation .....	80
System Clock Frequency Changes .....	80
PWM Period .....	79
Setup for PWM Operation .....	80
Timer Resources .....	75
CCP. <i>See</i> Capture/Compare/PWM (CCP)	
CCP1CON Register .....	75
Clock Sources	
External Modes .....	21
EC .....	21
HS .....	22
LP .....	22
OST .....	21
RC .....	23
XT .....	22
Internal Modes .....	23
Frequency Selection .....	25
HFINTOSC .....	23
INTOSC .....	23
INTOSCIO .....	23
LFINTOSC .....	25
Clock Switching .....	27
Code Examples	
A/D Conversion .....	64
Assigning Prescaler to Timer0 .....	42
Assigning Prescaler to WDT .....	42
Changing Between Capture Prescalers .....	76
Data EEPROM Read .....	73
Data EEPROM Write .....	73

# PIC12F683

Indirect Addressing .....	18
Initializing GPIO .....	31
Saving STATUS and W Registers in RAM .....	95
Ultra Low-Power Wake-up Initialization .....	35
Write Verify .....	73
Code Protection .....	99
Comparator .....	51
C2OUT as T1 Gate .....	57
Configurations .....	53
I/O Operating Modes .....	53
Interrupts .....	55
Operation .....	51, 54
Operation During Sleep .....	56
Response Time .....	54
Synchronizing COUT w/Timer1 .....	57
Comparator Module .....	
Associated registers .....	59
Comparator Voltage Reference (CVREF) .....	
Response Time .....	54
Comparator Voltage Reference (CVREF) .....	58
Effects of a Reset .....	56
Specifications .....	132
Comparators .....	
C2OUT as T1 Gate .....	45
Effects of a Reset .....	56
Specifications .....	132
Compare Module. See Capture/Compare/PWM (CCP) .....	
CONFIG Register .....	84
Configuration Bits .....	83
CPU Features .....	83
Customer Change Notification Service .....	171
Customer Notification Service .....	171
Customer Support .....	171
<b>D</b> .....	
Data EEPROM Memory .....	
Associated Registers .....	74
Code Protection .....	71, 74
Data Memory Organization .....	7
Map of the PIC12F683 .....	8
DC and AC Characteristics .....	
Graphs and Tables .....	137
DC Characteristics .....	
Extended and Industrial .....	121
Industrial and Extended .....	117
Development Support .....	111
Device Overview .....	5
<b>E</b> .....	
EEADR Register .....	71
EECON1 Register .....	72
EECON2 Register .....	72
EEDAT Register .....	71
EEPROM Data Memory .....	
Avoiding Spurious Write .....	74
Reading .....	73
Write Verify .....	73
Writing .....	73
Effects of Reset .....	
PWM mode .....	80
Electrical Specifications .....	115
Enhanced Capture/Compare/PWM (ECCP) .....	
Specifications .....	131
Errata .....	3

<b>F</b> .....	
Fail-Safe Clock Monitor .....	29
Fail-Safe Condition Clearing .....	29
Fail-Safe Detection .....	29
Fail-Safe Operation .....	29
Reset or Wake-up from Sleep .....	29
Firmware Instructions .....	101
Fuses. See Configuration Bits .....	
<b>G</b> .....	
General Purpose Register File .....	8
GPIO .....	31
Additional Pin Functions .....	32
ANSEL Register .....	32
Interrupt-on-Change .....	32
Ultra Low-Power Wake-up .....	32, 35
Weak Pull-up .....	32
Associated Registers .....	39
GP0 .....	36
GP1 .....	37
GP2 .....	37
GP3 .....	38
GP4 .....	38
GP5 .....	39
Pin Descriptions and Diagrams .....	36
Specifications .....	127
GPIO Register .....	31
<b>I</b> .....	
ID Locations .....	99
In-Circuit Debugger .....	100
In-Circuit Serial Programming (ICSP) .....	100
Indirect Addressing, INDF and FSR Registers .....	18
Instruction Format .....	101
Instruction Set .....	101
ADDLW .....	103
ADDWF .....	103
ANDLW .....	103
ANDWF .....	103
BCF .....	103
BSF .....	103
BTFSC .....	103
BTFSS .....	104
CALL .....	104
CLRf .....	104
CLRW .....	104
CLRWDt .....	104
COMF .....	104
DECF .....	104
DECFSZ .....	105
GOTO .....	105
INCF .....	105
INCFSZ .....	105
IORLW .....	105
IORWF .....	105
MOVF .....	106
MOVLW .....	106
MOVWF .....	106
NOP .....	106
RETFIE .....	107
RETLW .....	107
RETURN .....	107
RLF .....	108
RRF .....	108
SLEEP .....	108

SUBLW .....	108
SUBWF .....	109
SWAPF .....	109
XORLW .....	109
XORWF .....	109
INTCON Register .....	14
Internal Oscillator Block	
INTOSC	
Specifications .....	126, 127
Internal Sampling Switch (RSS) IMPEDANCE .....	67
Internet Address .....	171
Interrupts .....	92
ADC .....	64
Associated Registers .....	94
Comparator .....	55
Context Saving .....	95
Data EEPROM Memory Write .....	72
GP2/INT .....	92
GPIO Interrupt-on-change .....	93
Interrupt-on-Change .....	32
Timer0 .....	93
TMR1 .....	46
INTOSC Specifications .....	126, 127
IOC Register .....	34
<b>L</b>	
Load Conditions .....	124
<b>M</b>	
MCLR .....	86
Internal .....	86
Memory Organization	
Data EEPROM Memory .....	71
Microchip Internet Web Site .....	171
Migrating from other PIC Devices .....	165
MPLAB ASM30 Assembler, Linker, Librarian .....	112
MPLAB ICD 2 In-Circuit Debugger .....	113
MPLAB ICE 2000 High-Performance Universal	
In-Circuit Emulator .....	113
MPLAB ICE 4000 High-Performance Universal	
In-Circuit Emulator .....	113
MPLAB Integrated Development Environment Software ..	111
MPLAB PM3 Device Programmer .....	113
MPLINK Object Linker/MPLIB Object Librarian .....	112
<b>O</b>	
OPCODE Field Descriptions .....	101
OPTION Register .....	13, 43
OSCCON Register .....	20
Oscillator	
Associated registers .....	30, 48
Oscillator Module .....	19
EC .....	19
HFINTOSC .....	19
HS .....	19
INTOSC .....	19
INTOSCIO .....	19
LFINTOSC .....	19
LP .....	19
RC .....	19
RCIO .....	19
XT .....	19
Oscillator Parameters .....	126
Oscillator Specifications .....	125
Oscillator Start-up Timer (OST)	
Specifications .....	129

Oscillator Switching	
Fail-Safe Clock Monitor .....	29
Two-Speed Clock Start-up .....	27
OSCTUNE Register .....	24
<b>P</b>	
Packaging .....	159
Details .....	160
Marking .....	159
PCL and PCLATH .....	18
Computed GOTO .....	18
Stack .....	18
PCON Register .....	17, 88
PICSTART Plus Development Programmer .....	114
PIE1 Register .....	15
Pin Diagram .....	2
Pinout Descriptions	
PIC12F683 .....	6
PIR1 Register .....	16
Power-Down Mode (Sleep) .....	98
Power-On Reset (POR) .....	86
Power-up Timer (PWRT) .....	86
Specifications .....	129
Precision Internal Oscillator Parameters .....	127
Prescaler	
Shared WDT/Timer0 .....	42
Switching Prescaler Assignment .....	42
Program Memory Organization .....	7
Map and Stack for the PIC12F683 .....	7
Programming, Device Instructions .....	101
<b>R</b>	
Reader Response .....	172
Read-Modify-Write Operations .....	101
Registers	
ADCON0 (ADC Control 0) .....	65
ADRESH (ADC Result High) with ADFM = 0 .....	66
ADRESH (ADC Result High) with ADFM = 1 .....	66
ADRESL (ADC Result Low) with ADFM = 0 .....	66
ADRESL (ADC Result Low) with ADFM = 1 .....	66
ANSEL (Analog Select) .....	33
CCP1CON (CCP1 Control) .....	75
CMCON0 (Comparator Control) Register .....	56
CMCON1 (Comparator Control) Register .....	57
CONFIG (Configuration Word) .....	84
EEADR (EEPROM Address) .....	71
EECON1 (EEPROM Control 1) .....	72
EECON2 (EEPROM Control 2) .....	72
EEDAT (EEPROM Data) .....	71
GPIO .....	31
INTCON (Interrupt Control) .....	14
IOC (Interrupt-on-Change GPIO) .....	34
OPTION_REG (OPTION) .....	13, 43
OSCCON (Oscillator Control) .....	20
OSCTUNE (Oscillator Tuning) .....	24
PCON (Power Control Register) .....	17
PCON (Power Control) .....	88
PIE1 (Peripheral Interrupt Enable 1) .....	15
PIR1 (Peripheral Interrupt Register 1) .....	16
Reset Values .....	90
Reset Values (Special Registers) .....	91
STATUS .....	12
T1CON .....	47
T2CON .....	50
TRISIO (Tri-State GPIO) .....	32
VRCON (Voltage Reference Control) .....	58

# PIC12F683

WDTCN (Watchdog Timer Control).....	97	INT Pin Interrupt .....	94
WPU (Weak Pull-Up GPIO) .....	34	Internal Oscillator Switch Timing .....	26
Resets .....	85	Reset, WDT, OST and Power-up Timer .....	128
Brown-out Reset (BOR) .....	85	Time-out Sequence on Power-up (Delayed MCLR) ...	89
MCLR Reset, Normal Operation .....	85	Time-out Sequence on Power-up (MCLR with VDD) ..	89
MCLR Reset, Sleep .....	85	Timer0 and Timer1 External Clock .....	130
Power-on Reset (POR) .....	85	Timer1 Incrementing Edge .....	46
WDT Reset, Normal Operation .....	85	Two Speed Start-up.....	28
WDT Reset, Sleep .....	85	Wake-up from Sleep Through Interrupt .....	99
Revision History .....	165	Timing Parameter Symbology .....	124
<b>S</b>		TRISIO Register .....	32
Sleep .....		Two-Speed Clock Start-up Mode.....	27
Power-Down Mode .....	98	<b>U</b>	
Wake-up .....	98	Ultra Low-Power Wake-up .....	32, 35
Wake-up Using Interrupts .....	98	<b>V</b>	
Software Simulator (MPLAB SIM) .....	112	Voltage Reference. See Comparator Voltage	
Special Event Trigger .....	64	Reference (CVREF)	
Special Function Registers .....	8	Voltage References	
STATUS Register .....	12	Associated registers .....	59
<b>T</b>		VREF. SEE ADC Reference Voltage	
T1CON Register.....	47	<b>W</b>	
T2CON Register.....	50	Wake-up Using Interrupts .....	98
Thermal Considerations .....	123	Watchdog Timer (WDT).....	96
Time-out Sequence .....	88	Associated Registers .....	97
Timer0 .....	41	Clock Source .....	96
Associated Registers .....	43	Modes .....	96
External Clock .....	42	Period .....	96
Interrupt.....	13, 43	Specifications .....	129
Operation .....	41, 44	WDTCN Register .....	97
Specifications .....	130	WPU Register .....	34
T0CKI .....	42	WWW Address .....	171
Timer1 .....	44	WWW, On-Line Support .....	3
Associated registers.....	48		
Asynchronous Counter Mode .....	45		
Reading and Writing .....	45		
Interrupt.....	46		
Modes of Operation .....	44		
Operation During Sleep .....	46		
Oscillator .....	45		
Prescaler .....	45		
Specifications .....	130		
Timer1 Gate			
Inverting Gate .....	45		
Selecting Source .....	45, 57		
Synchronizing COUT w/Timer1 .....	57		
TMR1H Register .....	44		
TMR1L Register .....	44		
Timer2 .....			
Associated registers.....	50		
Timers			
Timer1			
T1CON .....	47		
Timer2			
T2CON .....	50		
Timing Diagrams			
A/D Conversion .....	135		
A/D Conversion (Sleep Mode) .....	135		
Brown-out Reset (BOR) .....	128		
Brown-out Reset Situations .....	87		
CLKOUT and I/O .....	127		
Clock Timing .....	125		
Comparator Output .....	51		
Enhanced Capture/Compare/PWM (ECCP) .....	131		
Fail-Safe Clock Monitor (FSCM) .....	30		

## THE MICROCHIP WEB SITE

Microchip provides online support via our WWW site at [www.microchip.com](http://www.microchip.com). This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

## CUSTOMER CHANGE NOTIFICATION SERVICE

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip web site at [www.microchip.com](http://www.microchip.com), click on Customer Change Notification and follow the registration instructions.

## CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support
- Development Systems Information Line

Customers should contact their distributor, representative or field application engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

**Technical support is available through the web site at: <http://support.microchip.com>**

## READER RESPONSE

It is our intention to provide you with the best documentation possible to ensure successful use of your Microchip product. If you wish to provide your comments on organization, clarity, subject matter, and ways in which our documentation can better serve you, please FAX your comments to the Technical Publications Manager at (480) 792-4150.

Please list the following information, and use this outline to provide us with your comments about this document.

To: Technical Publications Manager  
RE: Reader Response  
Total Pages Sent \_\_\_\_\_  
From: Name \_\_\_\_\_  
Company \_\_\_\_\_  
Address \_\_\_\_\_  
City / State / ZIP / Country \_\_\_\_\_  
Telephone: (\_\_\_\_\_) \_\_\_\_\_ - \_\_\_\_\_ FAX: (\_\_\_\_\_) \_\_\_\_\_ - \_\_\_\_\_

Application (optional):

Would you like a reply? \_\_\_Y \_\_\_N

Device: PIC12F683

Literature Number: DS41211D

Questions:

1. What are the best features of this document?

---

---

2. How does this document meet your hardware and software development needs?

---

---

3. Do you find the organization of this document easy to follow? If not, why?

---

---

4. What additions to the document do you think would enhance the structure and subject?

---

---

5. What deletions from the document could be made without affecting the overall usefulness?

---

---

6. Is there any incorrect or misleading information (what and where)?

---

---

7. How would you improve this document?

---

---



PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

<u>PART NO.</u>	<u>X</u>	<u>/XX</u>	<u>XXX</u>
Device	Temperature Range	Package	Pattern
<b>Device:</b>	PIC12F683 <sup>(1)</sup> , PIC12F683T <sup>(2)</sup> VDD range 2.0V to 5.5V		
<b>Temperature Range:</b>	I = -40°C to +85°C(Industrial) E = -40°C to +125°C (Extended)		
<b>Package:</b>	P = Plastic DIP MD = Dual-Flat, No Leads (DFN-S, 4x4x0.9 mm) MF = Dual-Flat, No Leads (DFN-S, 6x5 mm) SN = 8-lead Small Outline (3.90 mm)		
<b>Pattern:</b>	3-digit Pattern Code for QTP (blank otherwise)		

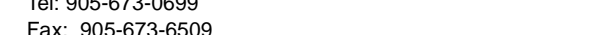
**Examples:**

a) PIC12F683-E/P 301 = Extended Temp., PDIP package, 20 MHz, QTP pattern #301

b) PIC12F683-I/SN = Industrial Temp., SOIC package, 20 MHz

**Note 1:** F = Standard Voltage Range  
LF = Wide Voltage Range

**2:** T = in tape and reel PLCC, and TQFP packages only.



Fax: 66-2-694-1350

Fax: 44-118-921-5820

## REFERENCIAS

- [1] Microchip, "Pic12f683 datasheet 8-pin flash-based 8-bit cmos micro-controllers," 2007.